2005

Design of a disturbance observer applied to a SCARA robotic arm

David L. Harlow
San Jose State University

DESIGN OF A DISTURBANCE OBSERVER APPLIED TO A SCARA ROBOTIC ARM

A Thesis

Presented to

The Faculty of

The Department of Mechanical and Aerospace Engineering

San Jose State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

David L. Harlow

May 2005

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

APPROVED FOR THE DEPARTMENT OF

MECHANICAL AND AEROSPACE ENGINEERING

Dr. Ji C. Wang

Dr. Burford J. Furman

Dr. Alain C. Carrier, Lockheed-Martin

APPROVED FOR THE UNIVERSITY

ABSTRACT

DESIGN OF A DISTURBANCE OBSERVER
APPLIED TO A SCARA ROBOTIC ARM

by David L. Harlow

Many applications in motion control require continuous improvement in basic performance (higher accuracy and higher acceleration) without the higher cost generally associated with such improvements. This is especially difficult in the presence of internal and external disturbances. The addition of a Disturbance Observer (DOB) to a typical motion control feedback loop can significantly and economically improve system performance without requiring any additional sensors. Moreover, the additional computational burden is very light. The DOB algorithms presented in this thesis are developed in continuous time and converted to discrete time using a trapezoidal algorithm with prewarping at the gain crossover frequency. Four DOB algorithms are implemented on the elbow link of an existing Selectively Compliant Assembly Robotic Arm (SCARA). Experimental results demonstrate the effect of DOB order on performance using second, third, and fourth order DOB algorithms.

Acknowledgements

First and foremost I wish to thank Dr. J.C. Wang for the use of the SJSU SCARA test facility and the Controls Laboratory. I am very grateful for all of his help, expertise, and patience throughout the course of this project. This project helped me to bridge the gap between control theory and the implementation of that theory in hands-on application.

I also wish to thank Dr. B. Furman for his helpful suggestions and questions, which contributed to the strength and depth of this thesis. Many of the skills acquired in his Mechatronics Lab class proved invaluable in this endeavor.

I thank Dr. A. Carrier for his many hours of personal attention both in the lab and in review of my documentation. Much of the success of this project is due to his recognition of the problems I experienced and guidance along the way.

I am eternally grateful to Dr. F. Barez for allowing me to complete this thesis on a schedule not frequently permitted in the Mechanical Engineering Department.

Lastly I wish to thank my mother, family, friends from church, and coworkers whose confidence in my abilities frequently exceeded my own. In addition, their sacrifices during the course of these several years while pursuing this degree has made this all come to a successful conclusion.

Table of Contents

List of Tables viii

List of Figures ix

List of Symbols xii

Chapter 1 Introduction 1

1.1 Problem Statement 3

1.2 Present Approach 4

1.3 Previous Studies Using the SJSU SCARA 5

1.4 Coordinate Frames 7

Chapter 2 Control Loop Design 11

2.1 Outer Control Loop Design 11

2.1.1 Plant Model Derivation 12

2.1.2 Plant Transfer Function 24

2.1.3 Tuning Outer Control Loop Gains 25

2.2 Disturbance Observer Design 28

2.2.1 Introduction to the Disturbance Observer 29

2.2.2 Real System Model 31

2.2.3 The Nominal Model 32

2.2.4 New Real System Model 34

2.2.5 Estimated Disturbance Torque 35

2.3 Characteristics of the Disturbance Observer 38
2.3.1 Q(S) Filters 40
2.3.2 Q(S) Filter Cutoff Frequency 43
2.3.3 Open And Closed Loop Transfer Functions And Stability 48
2.3.4 Open Loop Results 50

2.4 Modeling and Simulation 51
2.4.1 Continuous Closed Loop Simulation Model 52
2.4.2 Continuous Closed Loop Simulation Results 56
2.4.3 Discrete Closed Loop Simulation Model 60
2.4.4 Discrete Closed Loop Simulation Results 65

Chapter 3 Implementation 73

3.1 Scara Test Setup 74

3.2 Discrete Closed Loop Implementation Model 77

3.3 Control Desk Experiment Design 83

3.4 Test Results 85

Chapter 4 Conclusions 94

4.1 Recommendations 95

References 96

Appendices 99

Appendix A: SJSU Scara Parameters 100

Appendix B: Matlab m-Files 101

Appendix C: Simulation Plots 107

Appendix D: dSpace© and Control Desk 114

List of Tables

Table 2.1: Coefficients of Parameters $Q_{31D}(s)$ and $Q_{42D}(s)$ 43

Table 2.2: Relationship between Alpha vs. GM and PM for Discrete Time 70

Table 3.1: Test Results 92

List of Figures

Figure 1.1: Typical SCARA Robot Configuration 6

Figure 1.2: SCARA Coordinates and Geometrical Parameters 8

Figure 2.1: Block Diagram of a Typical SCARA Outer Control Loop 12

Figure 2.2: Schematic of SCARA Motors, Gears, and Links 13

Figure 2.3: Open Loop Voltage Sweeps for Friction Magnitudes 22

Figure 2.4: The Disturbance Observer Block Diagram 30

Figure 2.5: Equivalent Disturbance Observer Block Diagram 38

Figure 2.6: Frequency Response of the $Q_{20}(s)$ Filter (f_c = 57 Hz) 44

Figure 2.7: Comparison of the Order of Q(s) in Continuous Time 46

Figure 2.8: Closed Loop Model 46

Figure 2.9: Disturbance Observer Loop Showing α Effect 50

Figure 2.10: Open Loop Gain and Phase Margin – Continuous Time 51

Figure 2.11: Continuous Matlab / Simulink Model for the SJSU SCARA 53

Figure 2.12: Continuous Matlab / Simulink Plant Model for SJSU SCARA 54

Figure 2.13: Disturbance Observer Model for Simulation 56

Figure 2.14a: Continuous Closed-Loop Response 57
DOB Off and Disturbance Off

Figure 2.14b: Continuous Closed-Loop Response 58
DOB Off and Disturbance On

Figure 2.15a: Continuous Closed-Loop Response 59
DOB and Disturbance On
$Q_{20}(s)$ Filter Algorithm (Kaneko DOB)

Figure 2.15b: Continuous Closed-Loop Response 60
DOB and Disturbance On
$Q_{20}(s)$ Filter DOB

Figure 2.16: Discrete Matlab / Simulink Model for the SJSU SCARA 61

Figure 2.17: Bode Plot for $Q_{31}(s)$ DOB 65

Figure 2.18a: Discrete Closed-Loop Response 66
DOB and Disturbance Off

Figure 2.18b: Discrete Closed-Loop Response 67
DOB Off and Disturbance On

Figure 2.19: Discrete Time Simulation 68
$Q_{20}(s)$ Filter DOB (Kaneko)
f_c = 37Hz; α = 0.90; prewarp at 17.645 rad/sec

Figure 2.20: Discrete Time Simulation 68
$Q_{20}(s)$ Filter DOB with damping
f_c = 37Hz; α = 0.90; prewarp at 17.645 rad/sec

Figure 2.21: Discrete Time Simulation 69
$Q_{31D}(s)$ DOB
f_c = 44 Hz; α = 0.90; prewarp at 17.645 rad/sec

Figure 2.22: Open Loop Gain and Phase Margin – Discrete Time 71
(Clockwise from upper left, α = 1.0, 0.95, 0.90, 0.85)

Figure 2.23: Disturbance Observer Algorithm Robustness 72

Figure 3.1: SJSU SCARA 73

Figure 3.2: SJSU SCARA Test Setup 75

Figure 3.3: dSpace© ds1102 DSP Controller Board 76

Figure 3.4: Matlab / Simulink Model for Elbow Robotic Arm 77

Figure 3.5: Simulink Model to Compare Encoder Velocity vs. Velocity 79

Figure 3.6: Raw test data showing results of the Encoder Test 80

Figure 3.7: Encoder Digital Resolution 82

Figure 3.8: dSpace© ControlDesk© Experimentation Layout 84

Figure 3.9: dSpace© Run with Disturbance Off and DOB Off 86

Figure 3.10: dSpace© Run with Disturbance On and DOB Off 87

Figure 3.11: dSpace© Run with Q_{20}, no Damping (Kaneko Algorithm) 88
f_c = 60 Hz; α = 0.90

Figure 3.12: dSpace© Run with Q_{20} with Damping 89
f_c = 60 Hz; α = 0.90

Figure 3.12: dSpace© Run with Q_{31D} with Damping 90
f_c = 60 Hz; α = 0.90

Figure 3.14: dSpace© Run with Q_{42D} with Damping 91
f_c = 60 Hz; α = 0.90

List of Symbols

Symbol	Description	Units
α	DOB tuning parameter	-
α_n	Q_{20} tuning parameters, cutoff frequency	-
B_{EFF}	Viscous friction damping, elbow effective	oz_f-in-s^2
B_m	Viscous friction damping, motor	oz_f-in-s/rad
B_n	Viscous friction damping, elbow, nominal	oz_f-in-s/rad
$C(\Theta, \ddot{\Theta})$	Coriolis and centrifugal torque terms	oz_f-in
$F(\Theta, \dot{\Theta})$	Friction torque terms	oz_f-in
f_c	Frequency, cutoff	Hz
F_{EFF}	Friction torque, elbow effective	oz_f-in
F_m	Coulomb friction torque, motor	oz_f-in
g_1, g_2	Q_{20} tuning parameters, cutoff frequency	-
G_n, $G_n(s)$	Nominal model, plant, general	rad/oz_f-in
G_{n1}, $G_{n1}(s)$	Nominal model, plant, no damping (Kaneko)	rad/oz_f-in
G_{n2}, $G_{n2}(s)$	Nominal model, plant, with damping	rad/oz_f-in
G_p, $G_p(s)$	Real model, plant	rad/oz_f-in
I	Inertia, elbow (about cg)	lb_f-in-s^2
I_d	Current, disturbance, estimate	oz_f-in
$\hat{I}_d$, $\hat{I}_d(s)$	Current, disturbance, estimate	oz_f-in
I_m	Current, motor coil	A

Symbol	Description	Units
I_{REF}	Current, reference	A
L_m	Inductance, motor coil	H
J_E	Inertia, rotational, equiv	oz_f-in-s^2
J_G	Inertia, rotational, gear	oz_f-in-s^2
J_m	Inertia, rotational, motor rotor	oz_f-in-s^2
J_{EFF}	Inertia, rotational, elbow effective	oz_f-in-s^2
J_n	Inertia, rotational, nominal elbow	oz_f-in-s^2
k_c	Q_{31D}, Q_{42D} tuning parameter	-
K_p	Gain, controller, proportional	oz_f-in//rad
K_d	Gain, controller, derivative	oz_f-in-sec//rad
K_m	Gain, oscillation	oz_f-in//rad
K_t	Torque constant, motor	oz_f-in//A
K_e	Back EMF constant, motor	V-sec//rad
L	Length, elbow (pivot to end)	in
L_c	Length, elbow (pivot to cg)	in
m	Mass, elbow	lbm
N, N(s)	Noise, measurement	-
N_g	Gear ratio	-
Q, Q(s)	Q(s) Filter	-
Θ^*, $\Theta^*(s)$	Angular position, elbow, desired	rad
Θ, $\Theta(s)$	Angular position, elbow, actual	rad

Symbol	Description	Units
Θ_e, $\Theta_e(s)$	Angular position error, elbow	rad
$\dot{\Theta}$, $\dot{\Theta}(s)$	Angular velocity, elbow, actual	rad/sec
$\ddot{\Theta}$, $\ddot{\Theta}(s)$	Angular acceleration, elbow, actual	rad/sec^2
$\hat{\Theta}$, $\hat{\Theta}(s)$	Angular position, elbow, measured	rad
$\dot{\overline{\Theta}}$, $\dot{\overline{\Theta}}(s)$	Angular velocity, elbow, measured	rad
θ	Angular position, elbow, actual	rad
θ_e	Angular position, elbow, error	rad
$\dot{\theta}$	Angular velocity, elbow, actual	rad/sec
$\ddot{\theta}$	Angular acceleration, elbow, actual	rad/sec^2
θ_m	Angular position, motor	rad
$\dot{\theta}_m$	Angular velocity, motor	rad/s
$\ddot{\theta}_m$	Angular acceleration, motor	rad/s^2
R_m	Resistance, motor coil	Ohms
s	Laplace Transform differentiator	sec^{-1}
τ	Q_{31D}, Q_{42D} tuning parameter, cutoff frequency	-
τ_v	Time constant, velocity filter	-
T	Torque, elbow	oz$_f$-in
$\tilde{T}$, $\tilde{T}(s)$	Torque, disturbance, derived	oz$_f$-in
T_C, $T_C(s)$	Torque, control	oz$_f$-in

Symbol	Description	Units
T_d	Torque, disturbance	oz_f-in
$\hat{T}_d$	Torque, disturbance, estimate	oz_f-in
T_E	Torque, equivalent	oz_f-in
T_{EXT}	Torque, disturbance	oz_f-in
T_G	Torque, gear	oz_f-in
T_L	Torque, load, reflected to motor	oz_f-in
T_m	Torque, motor	oz_f-in
T_{REF}	Torque, reference (PD control)	oz_f-in
T_s	Time, digital sampling	sec
V_{in}	voltage, input, motor	Volts
V_e	Voltage, back EMF	Volts
ω_c	frequency, cutoff	rad/sec
ω_N	frequency, Nyquist	rad/sec
ω_o	frequency, prewarp	rad/sec
ω_s	frequency, sampling	rad/sec
ζ	damping ratio	-
z	z-Transform, delay	sec

Chapter 1

Introduction

Advances in motion control applications have been steadily increasing for more than a decade. These advances are driven by a general industry demand for continuous improvement in basic performance (higher accuracy and higher acceleration). Motion control applications include very small systems such as the Compact Disk (CD), the Digital Versatile Disk or Digital Video Disk (DVD), Digital Still Cameras (DSC), and computer disk drives. Applications also include large systems such as satellite dish antenna position and tracking control, aircraft control surfaces, and industrial process control machines. In between these are robotic manipulators for assembly and manufacturing of various products, welding, inspection, etc.

At present, data storage capacities for the DVD range from a minimum of 4.7 GB to 17 GB and data access rates of 600 KBps (kilobytes per second) to 1.3 MBps (Megabytes per second), (Webopedia, n.d., DVD). Thus the DVD represents a tenfold increase in data storage capacity over the CD at 650 MB (Megabytes). As storage capacity increases the number of concentric data tracks must also increase for the same surface area or size of disk. Therefore, to accommodate the smaller track size, the position accuracy of the disk read-write head must also increase. Moreover, to accommodate increased data rates, the speed of the disk is increased, requiring higher acceleration of the read-write head.

Disturbances arise from the motion of the unit (shock and vibration) and also from the high rotational speed of the motor.

A second example of a motion control application is a large dish antenna that must be pointed at a satellite or other transmitting station to receive a signal. Azimuth and elevation actuators are used to control the antenna alignment to lock onto a stationary target or to track a moving target. Disturbances may be generated by the wind, ground movement, electrical interference, climate changes, etc.

As final example of a motion control application, robotic manipulators are devices which have two or more controlled linkages used to perform various repetitive tasks. These include electrical and mechanical assembly, x-ray or photographic inspection, welding in the automotive industry, etc. Disturbances may be generated externally as a direct force on the robotic arm, shock and vibration. Additionally, variance in inertia, as when a part is retrieved or deposited, can be significant enough to cause a change in kinematics and therefore be considered a disturbance.

The performance of Classical Control algorithms have worked well and will continue to do so. However, the performance requirements for some applications are beyond the capability of control laws such as the Proportional-Derivative (PD) or Proportional-Integral-Derivative (PID) algorithms alone, especially in the presence of internal or external disturbances.

1.1 Problem Statement

It is desirable to improve the performance of the various motion control applications described above. Recent advances in microprocessor technology make it possible for advanced controller algorithms to be implemented, where previously these controllers barely had programmable space for the conventional PID algorithm. As these microprocessors gradually generated more programmable space and enabled more complicated algorithms with smaller sampling times, higher performance and flexibility have been realized without additional cost (Ohnishi, 1996).

The disturbance observer (DOB) is one example of a model-based advanced control algorithm. The DOB was originally proposed in 1987 by Kouhei Ohnishi (White, 1999). It is an auxiliary controller to the conventional (PD or PID) controllers. The DOB algorithm is designed to produce robust plant behavior by canceling or suppressing the effects of disturbances in a specified frequency range. The algorithm consists of a nominal model of the plant dynamics, and a filter. The actual output of the plant is processed using the inverse of this nominal model, the control input, and the measured feedback to produce an estimate of the disturbance which is feed-forwarded to cancel the actual disturbance as it enters the plant. The DOB algorithm is simple and therefore only a light additional computational burden is placed on the microprocessor.

Most of the motion control applications previously described consist of mass-produced components. This implies that inertias and other component performance characteristics will vary from device to device to some degree. Nominal values of these properties can be estimated via statistical average from a small sample of devices. The actual value of these performance parameters for any particular unit is equal to the nominal plus or minus some variation. Since the nominal model consists only of the nominal value of these characteristics, a portion of the disturbance estimate will be due to these variations in addition to other disturbances whose dynamics are not included in the nominal model, thus driving the plant to behave as a nominal plant (theoretically).

1.2 Present Approach

The DOB algorithms presented in this thesis are developed using a continuous time simulation model. Once acceptable performance has been obtained, the continuous time model is converted to discrete time. The DOB algorithms are converted to discrete time using a binomial algorithm with prewarping at the gain crossover frequency. This process will ensure that the performance of the discrete transfer functions will closely match their counterparts in continuous time for low frequencies (up to the Nyquist frequency of the discrete closed loop control system).

The platform selected to demonstrate the DOB algorithms in this thesis is the elbow link of the SJSU Selectively Compliant Assembly Robotic Arm (SCARA). The SCARA is the simplest of the industrial robotic manipulators. It generally consists of two links and a manipulator. Figure 1.1 illustrates the typical arrangement of the SCARA, with a typical workspace coverage shown in crosshatch, (Craig, 1989, p.269). The SJSU SCARA control actuator for each arm consists of a brush-type DC motor with an incremental encoder for angular feedback, planetary gear head, and an amplifier. The hardware is welded aluminum cylinders and channels to make up the two links. The two joints each employ roller bearings. Unfortunately, at the time of this thesis the shoulder motor is behaving erratically. Thus the control implementation of the DOB algorithms is limited to the elbow only. The two degree of freedom SCARA would have been a better platform to demonstrate the effectiveness of the DOB algorithm, in part due to the coupled dynamics between the shoulder and elbow links. Additionally, no practical means to lock the shoulder link in place was found, therefore this link was held in place manually during each run.

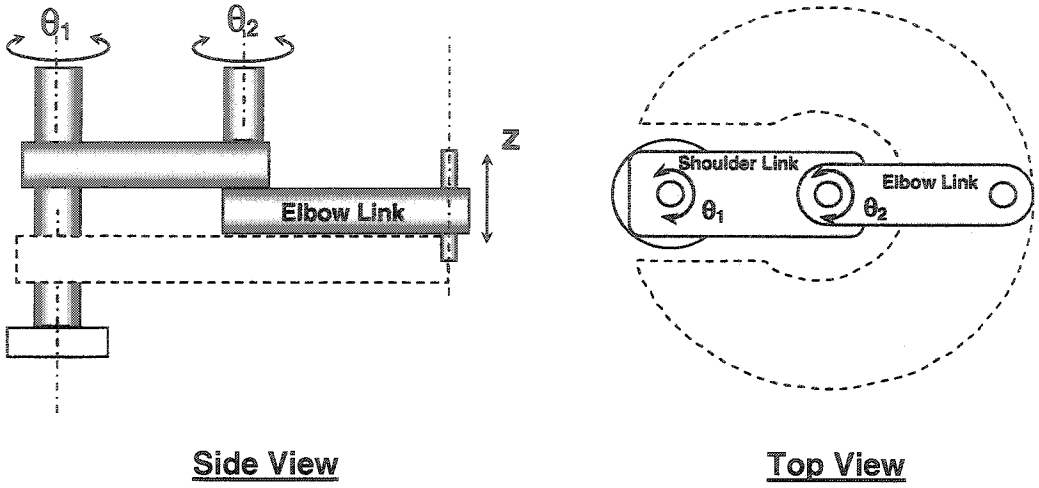

Figure 1.1: Typical SCARA Robot Configuration

Typical disk-shaped workspace coverage is shown in crosshatch. Labels indicate joint motions. This configuration is best suited to planar tasks.

1.3 Previous Studies Using the SJSU SCARA

Two Master's Projects have been performed using the SJSU SCARA. The first (Tai, 2002) implemented a modified PD or computed torque controller for track following a linear path using the robotic arm and a dSpace© DS1102 Digital Signal Processor (DSP) controller. The trajectory of the path was given in Cartesian Space and converted to Joint Space using inverse kinematics.

The second Master's Project using the SJSU SCARA (Dang, 2003) implemented a fourth order disturbance observer on the elbow portion of the robotic arm and also used the dSpace DS1102 DSP controller. A modified PD controller and DOB were used to reject a simulated step input disturbance torque and torque due to inertia change.

1.4 Coordinate Frames

Two common coordinate frames for SCARA robotics are called Joint space and Cartesian space. The latter is also referred to as operational space or task space. Joint space is the coordinate frame in which the dynamics of the manipulator (tip location) is determined via kinematics from the joint angles (θ_1 and θ_2) of each link (Figure 1.2). Let θ_1 represent the angle the shoulder makes with the inertially-fixed X-axis. Let θ_2 represent the angle the elbow link makes relative to the shoulder link.

Cartesian space is a coordinate frame in which the location of the tip is described by a set of orthogonal (x, y) coordinates locating the tip, usually with the origin at the base (as shown in Figure 1.2). Both coordinate frames come with advantages and disadvantages. The two links will hereafter be referred to as the shoulder link (driven by the shoulder motor and designated by its length, L_1) and the elbow link (driven by the elbow motor and designated by its length, L_2). Let L_{c1} and L_{c2} represent the length from each link pivot point to its center of mass. The

numerical values of these and other parameters for the SJSU SCARA are listed in Appendix A.

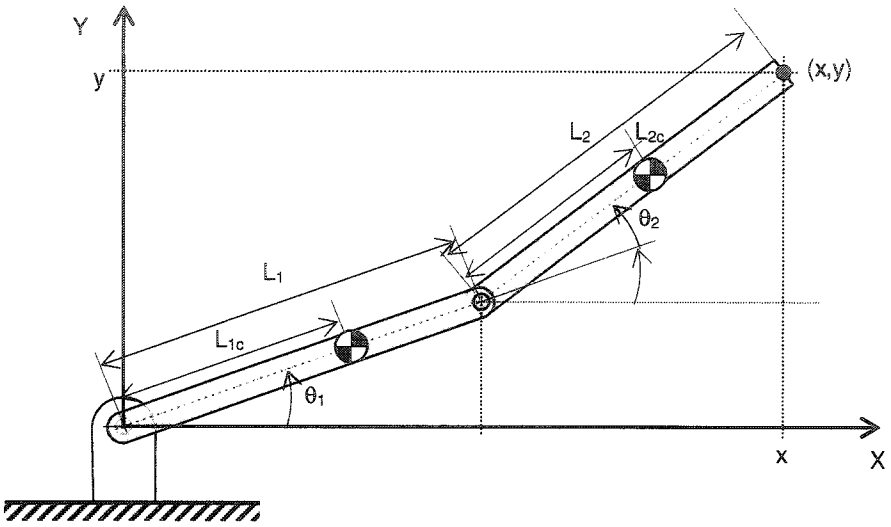

Figure 1.2: SCARA Coordinates and Geometrical Parameters

The SCARA system showing Cartesian and Joint Space coordinate frames and the relationship between them. Labels indicate geometrical parameters of the SCARA system. Programmed trajectories are frequently more readily given in Cartesian coordinates of the tip making it necessary to convert to Joint Space.

The desired trajectory is (by definition) a time history of the tip path given in tip coordinates. The desired trajectory for a given task can be summarized as a set of movements easily described in Cartesian space (a series of straight lines in x, y coordinates), but not so easily in joint space (joint angles corresponding to the straight lines).

Control in Cartesian Space has the benefit of controlling on the desired parameters; i.e., the desired (x,y) coordinates of the tip. Moreover, the error between the actual tip coordinates and the desired tip coordinates is driven to zero by the controller. However, the robot is typically controlled by DC motors at the shoulder and elbow joints. Commands to the motors (torques) and position feedbacks (joint angles) are inherently made in Joint Space regardless of the coordinate frame chosen for the outer control loop. Therefore control in Cartesian Space requires some geometrical conversions involving the Jacobian matrix for the links to be performed inside the control loop. This may be an issue depending on the capability of the DSP.

In Joint space control, the outer loop is closed around the joint angles. While the feedback measurements are made of the joint angles, the desired position must be in similar units of measure; i.e., desired joint angles. However, specifying the desired tip position in Cartesian space is more natural. A geometrical conversion must be made in this frame as well, but it can be performed outside the control loop. As such the inner loop is less math-intensive since conversion from Cartesian space (x,y) commands to Joint space (θ_1,θ_2) commands using various forms of the Jacobian matrix for the links can be performed outside the control loop. Some minor inaccuracy may be incurred, however, since the control loop is operating on the joint angles rather than the desired (x,y) coordinates of the tip.

By necessity, the coordinate frame chosen for this thesis is Joint space. Although any realistic trajectory requires both links functioning, the primary goal of this thesis is to develop, implement and compare DOB algorithms.

Chapter 2

Control Loop Design

The design of the control system takes place in two parts: 1. design of the outer control loop and, 2. design of the disturbance observer or inner control loop. The design of the outer control loop will begin with the derivation of the plant model. This design is followed by an analysis of the behavior of the control system.

2.1 Outer Control Loop Design

The outer control loop consists of the modified PD controller, motor power amplifier, SCARA plant model, and the sensor model (as shown in Figure 2.1). The reference torque, $T_{REF}(s)$, is the command signal from the controller. The saturation element represents saturation in the motor power drive electronics or amplifier, and has a gain with units of A/V. The disturbance torque, $T_d(s)$, is assumed to enter the plant as a direct torque on the controlled element. Therefore it is placed downstream of the saturation element in this diagram. In the laboratory implementation (and analysis which reflects the implementation), the disturbance is limited for safety concerns. The measured angular position is $\hat{\Theta}(s)$ which includes measurement (sensor) noise, N(s). The derived angular velocity, $\dot{\hat{\Theta}}(s)$, is

produced by averaging the measured angular position using a first order filter and then differentiating.

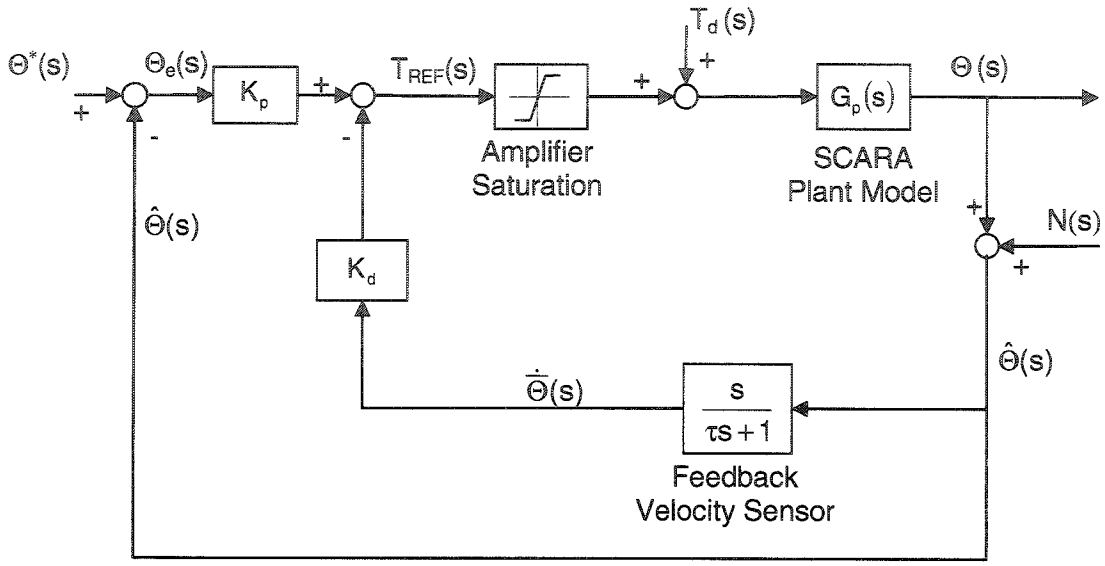

Figure 2.1: Block Diagram of a Typical SCARA Outer Control Loop

Typical SCARA outer control loop showing nomenclature used in this thesis.

2.1.1 Plant Model Derivation

The disturbance observer is implemented using only the elbow portion of the SCARA robot (as shown in Figure 2.2). The plant model describing the dynamics of the elbow link begins with its drive motor. Although the elbow link motor is mounted to the shoulder link, the shoulder link is assumed to be fixed in a

Newtonian Reference Frame. Therefore coupling torques from the shoulder link are eliminated.

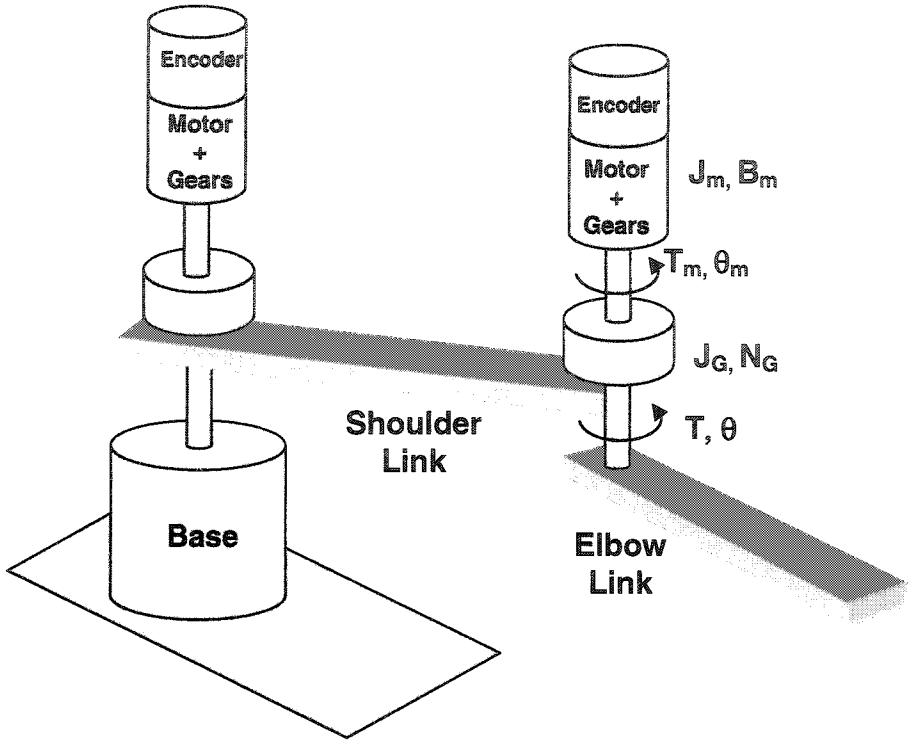

Figure 2.2: Schematic of SCARA Motors, Gears, and Links

The voltage in the DC motor coil windings is described by (2.1.1.-1):

$$V_m = L_m \cdot \frac{d}{dt} I_m + R_m \cdot I_m + V_e \qquad (2.1.1\text{-}1)$$

where:

V_m = input voltage to the motor coils, volts

V_e = back electro magnetic force (emf) voltage, volts

I_m = motor coil current, amps

R_m = motor coil resistance, ohms

L_m = motor coil inductance, henries

A motor control voltage (V_m^*) is created by the digital controller and sent to the motor control power amplifier. An amplifier gain (G_a) converts the control voltage to a control current (I_c). A current control amplifier is used to control the motor driving the elbow link. The electronics in this type of amplifier are designed to make the effect of the motor inductance and back electromotive force (emf) negligible. Therefore the motor coil current (I_m) is approximately equal to the control current by the design of the current amplifier.

The current amplifier equation can be written as 2.1.1-2a:

$$I_c = G_a \cdot V_m^* \qquad (2.1.1\text{-}2a)$$

where:

V_m^* = motor control voltage, volts

G_a = amplifier gain, A/V

I_c = motor control current, A

and the motor coil current is shown in 2.1.1-2b:

$$I_m \cong I_c \qquad (2.1.1\text{-}2b)$$

The proportional relationship between motor torque and coil current for a given motor is shown in Equation 2.1.1-3, where the motor torque constant, K_t, is given in units of torque/A such as oz_f-in/A:

$$T_m = K_t \cdot I_m \qquad (2.1.1\text{-}3)$$

where:

T_m = motor torque, oz_f-in

K_t = motor torque constant, oz_f-in/A

Newton's Law for the motor rotor gives the motor torque that must be generated to drive the elbow link is given by (2.1.1-4):

$$T_m = J_m \cdot \ddot{\theta}_m + B_m \cdot \dot{\theta}_m + F_m + T_L \qquad (2.1.1\text{-}4)$$

where:

J_m = rotational inertia of the motor rotor, oz_f-in-s^2

B_m = viscous friction damping in the motor bearings and magnetic field, oz_f-in-s/rad

F_m = Coulomb friction torque of the motor sliding or rolling parts, oz_f-in

T_L = load torque reflected to the motor rotor, oz_f-in

$\ddot{\theta}_m$ = motor angular acceleration, rad/s^2

$\dot{\theta}_m$ = motor angular velocity, rad/s

The combination of (2.1.1-3) and (2.1.1-4) yield a torque-balance for the motor rotor system, including the load as shown by equation 2.1.1-5:

$$K_t \cdot I_m = J_m \cdot \ddot{\theta}_m + B_m \cdot \dot{\theta}_m + F_m + T_L \qquad (2.1.1\text{-}5)$$

Since the elbow mechanism has a gearbox between the motor output gear and the elbow link (in this case a planetary gearbox), the motor will see only a small fraction of the load torque, $\frac{1}{N_G}$, where N_G is the gear ratio of the gearbox and is greater than one. As seen in (2.1.1-6), the load torque reflected to the motor is reduced by this gear ratio, and consequently, the speed which the motor must develop must be increased by the multiple of the gear ratio.

$$T_L = \frac{1}{N_G} \cdot (T_G + T) \qquad (2.1.1\text{-}6)$$

where:

T_G = gear torque, oz_f-in

T = elbow link torque, oz_f-in

N_G = gear ratio, (-)

Since the load torque is built up from a gearbox and linkage parts, let T_E represent the equivalent torque of these parts in such a way that T_E is related back to the load torque via Equation 2.1.1-7:

$$T_L = \frac{1}{N_G} \cdot (T_E) \tag{2.1.1-7}$$

where:

T_E = equivalent torque between the gear and elbow, oz_f-in

Employing Newton's Law, this time for the gearbox and link gives:

$$T_E = J_E \cdot \ddot{\theta} + B_E \cdot \dot{\theta} + F_E \cdot \text{sign}(\dot{\theta}) \tag{2.1.1-8}$$

where:

J_E = equivalent inertia between the gear and elbow, oz_f-in-s^2

B_E = equivalent damping between the gear and elbow, oz_f-in-s/rad

F_E = equivalent friction torque between the gear and elbow, oz_f-in

$\ddot{\theta}$ = elbow link angular acceleration, rad/s^2

$\dot{\theta}$ = elbow link angular velocity, rad/s

In Equation 2.1.1-8, J_E and B_E represent the lumped inertia and viscous damping, respectively, for the gearbox and linkage. The magnitude of the Coulomb friction for the gearbox and linkage is represented by F_E. Certain numerical details of the linkage properties must be known in order to compute the lumped inertia. Let J_G be the rotary inertia of the planetary gearbox, I is the moment of inertia of the elbow link about a vertical axis through its center of gravity (cg). The parallel axis theorem yields the inertia of the elbow link about its pivot point as seen in (2.1.1-9):

$$J_E = J_G + I + m \cdot L_c^2 \qquad (2.1.1\text{-}9)$$

where:

J_E = equivalent inertia between the gear and elbow, oz_f-in-s^2

I = elbow moment of inertia about its cg, oz_f-in-s^2

m = mass of the elbow link, lb_m

L_c = length, elbow pivot point to its cg, in

Lubrication in the sliding surfaces accounts for much or most of the viscous damping, B_E, in this mechanism. This parameter will be determined experimentally as will be discussed later. Substitute 2.1.1-7 and 2.1.1-8 and then substitute the resulting expression for T_L into (2.1.1-5) resulting in (2.1.1-10):

$$K_t \cdot I_m = J_m \cdot \ddot{\theta}_m + B_m \cdot \dot{\theta}_m + F_m \cdot \text{sign}(\dot{\theta}_m) + \frac{1}{N_G} \cdot (J_E \cdot \ddot{\theta} + B_E \cdot \dot{\theta} + F_E \cdot \text{sign}(\dot{\theta})) \quad (2.1.1\text{-}10)$$

Equation 2.1.1-10 contains θ terms representing both the motor dynamics as well as the elbow link dynamics. At this point it would be convenient to convert one to the other and combine terms. This can be done relatively easily using the relationship of 2.1.1-11 and its derivatives in 2.1.1-12 and 2.1.1-13.

$$\theta_m = N_G \cdot \theta \quad (2.1.1\text{-}11)$$

$$\dot{\theta}_m = N_G \cdot \dot{\theta} \quad (2.1.1\text{-}12)$$

$$\ddot{\theta}_m = N_G \cdot \ddot{\theta} \quad (2.1.1\text{-}13)$$

In this case, since the control system loop closure is on elbow link position, θ, and this parameter will be required in the controller. Equation 2.1.1-16 should be rewritten reflecting all motor (θ_m) terms to elbow link (θ) terms as in (2.1.1-14):

$$K_t \cdot I_m = J_m \cdot N_G \cdot \ddot{\theta} + B_m \cdot N_G \cdot \dot{\theta} + F_m \cdot \text{sign}(\dot{\theta}) + \frac{1}{N_G} \cdot (J_E \cdot \ddot{\theta} + B_E \cdot \dot{\theta} + F_E \cdot \text{sign}(\dot{\theta})) \quad (2.1.1\text{-}14)$$

Collecting like-terms of θ results in (2.1.1-15):

$$K_t \cdot I_m = \left(J_m \cdot N_G + \frac{J_E}{N_G} \right) \cdot \ddot{\theta} + \left(B_m \cdot N_G + \frac{B_E}{N_G} \right) \cdot \dot{\theta} + \left(F_m + \frac{F_E}{N_G} \right) \cdot \text{sign}(\dot{\theta}) \quad (2.1.1\text{-}15)$$

A little modification to equation 2.1.1-15 is appropriate to facilitate Matlab / Simulink modeling. To do this, create "effective" parameters for the constant terms on the right hand side of Equation 2.1.1-15:

$$J_{EFF} = J_m \cdot N_G + \frac{J_E}{N_G} \quad (2.1.1\text{-}16)$$

$$B_{EFF} = B_m \cdot N_G + \frac{B_E}{N_G} \quad (2.1.1\text{-}17)$$

$$F_{EFF} = \left(F_m + \frac{F_E}{N_G} \right) \quad (2.1.1\text{-}18)$$

where:

J_{EFF} = effective inertia for the motor, gear, and elbow link, $oz_f\text{-in-s}^2$

B_{EFF} = effective damping for the motor, gear, and elbow link, $oz_f\text{-in-s/rad}$

F_{EFF} = effective friction for the motor, gear, and elbow link, $oz_f\text{-in}$

Substituting (2.1.1-16), (2.1.1-17), and (2.1.1-18) into (2.1.1-15) results in equation (2.1.1-19):

$$K_t \cdot I_m = J_{EFF} \cdot \ddot{\theta} + B_{EFF} \cdot \dot{\theta} + F_{EFF} \cdot \text{sign}(\dot{\theta}) \qquad (2.1.1\text{-}19)$$

It was stated earlier that the effective viscous friction, B_{EFF}, and the effective Coulomb friction, F_{EFF}, terms are experimentally derived properties. This was accomplished in the lab first by running a static test, for the Coulomb friction. Initially the elbow joint is stationary in a static test (by definition) which makes all of the time dependent terms zero in (2.1.1-19), resulting in (2.1.1-20) (Tai, 2002):

$$K_t \cdot I_m = F_{EFF} \qquad (2.1.1\text{-}20)$$

The effective Coulomb friction was determined by parametrically decreasing the current from large values until motion was just detectable via the encoder. The resulting value for F_{EFF} is 1.632 oz_f-in, (Tai, 2002). This value of the Coulomb friction was confirmed by testing conducted as part of this thesis, although not to four significant figures. It is doubtful whether this degree of accuracy is justifiable. Figure 2.3 shows the parametric voltage sweeps. Each of the plots was time biased to zero at the start of the step input. The elbow link was initially placed so as to allow the largest possible stroke, nearly 360°, prior to each run. The last two or three runs (minimum voltage) displayed no perceptual motion

visually. The Coulomb friction value was determined using 2.2 volts, the amplifier gain of 0.077 A/V, and the motor torque constant of 9.6 oz_f-in/A resulting in a friction magnitude of 1.626 oz_f-in.

Similarly, the viscous friction is found using the Coulomb friction and a constant velocity portion from the above experiment. This eliminates the acceleration term and (2.1.1-19) results in (2.1.1-21):

$$K_t \cdot I_m = B_{EFF} \cdot \dot{\theta} + F_{EFF} \qquad (2.1.1\text{-}21)$$

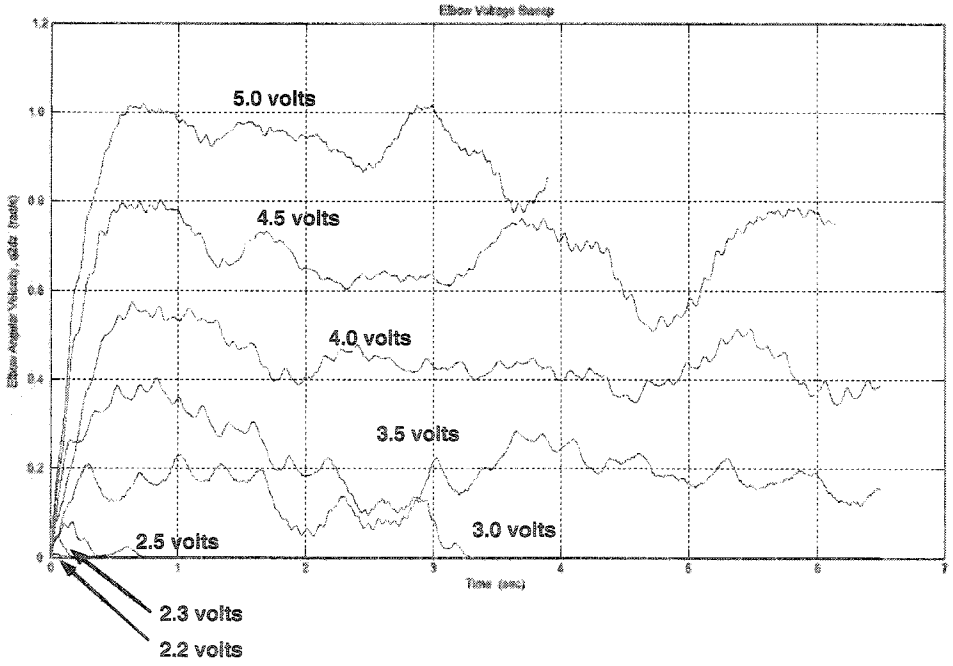

Figure 2.3: Open Loop Voltage Sweeps for Friction Magnitudes

Experimental data from the SJSU SCARA elbow link of angular velocity vs. time for a family of motor coil voltages.

As in the case of the Coulomb friction, viscous friction testing confirmed Ta-Jen's result. The 4.0 volt plot in Figure 2.3 yields the most constant velocity of any of the runs. An average velocity was determined for the time between 2.0 sec and the end of the test. Using Matlab, the mean angular velocity over this range is 0.4217 rad/sec. Equation 2.1.1-21 is easily solved for the effective viscous friction given the effective Coulomb friction from the previous test, and yields 3.142 oz_f-in-sec/rad. This value of the viscous friction is quite close to Ta-Jen's 3.438 oz_f-in-sec/rad. Although it makes little difference numerically, Ta-Jen's original friction measurements will be used for this thesis.

The effective inertia is derived from known properties of the robotic system. These properties are found in Appendix A. The effective inertia can be derived by substituting (2.1.1-9) into (2.1.1-16):

$$J_{EFF} = J_m \cdot N_G + \frac{J_G + I + m \cdot L_c^2}{N_G} \qquad (2.1.1\text{-}22)$$

Substitution of the respective parametric values from Appendix A into (2.1.1-22) results in J_{EFF} = 0.318 oz_f-in-s^2.

Finally, each of the derived and experimental parameters can be substituted into (2.1.1-19) resulting in the plant model.

$$9.6 \cdot I_m = 0.318 \cdot \ddot{\theta} + 3.438 \cdot \dot{\theta} + 1.632 \cdot \text{sign}(\dot{\theta}) \qquad (2.1.1\text{-}23)$$

Many different methods have been used to implement the Coulomb friction in the plant model (using a simulation language like Simulink). It is best to start with a very simple friction model and upgrade only if greater accuracy is necessary. The simplest Coulomb friction model is a magnitude multiplied by the "sign" function in Simulink. The function $\text{sign}(\dot{\theta})$ implies that only the sign (positive or negative) of the elbow link angular velocity is used to modify F_{EFF}.

2.1.2 Plant Transfer Function

The plant transfer function relates the plant output to its input, $\left(\frac{\Theta}{I_m}\right)$. Let **s** be the Laplace differentiator, d/dt; then from (2.1.1-23), temporarily disregarding the nonlinear friction term (since the Laplace transform is undefined for nonlinear terms), equation (2.1.2-1) is derived.

$$9.6 \cdot I_m = 0.318 \cdot \Theta \cdot s^2 + 3.438 \cdot \Theta \cdot s \qquad (2.1.2\text{-}1)$$

The friction term will however be used in the dynamic model of the robot system to test the performance of the disturbance observers prior to implementation.

Factoring (2.1.2-1):

$$9.6 \cdot I_m = \left(0.318 \cdot s^2 + 3.438 \cdot s\right) \cdot \Theta \qquad (2.1.2\text{-}2)$$

Equation 2.1.2-3 results in the plant transfer-function, $G_p(s)$:

$$G_p(s) \equiv \frac{\Theta(s)}{I_m(s)} = \frac{9.6}{0.318 \cdot s^2 + 3.438 \cdot s} \qquad (2.1.2\text{-}3)$$

2.1.3 Tuning Outer Control Loop Gains

The closed loop transfer function relates the actual (feedback) angular position of the elbow link to the desired position. The outer loop PD control gains are tuned independently of the DOB.

Let $\Theta^*(s)$ represent the desired angular position of the elbow link, and $\hat{\Theta}(s)$ the measured feedback angular position, the latter being defined by the sensor equation, (2.1.3-1). The angular velocity is derived from the measured feedback (2.1.3-2) and represented by $\dot{\hat{\Theta}}(s)$ as described at the opening of this chapter, and

shown graphically in the block diagram in Figure 2.1. Here, a time constant of 0.020 seconds was found to yield a close approximation of the actual velocity.

$$\hat{\Theta}(s) = \Theta(s) + N(s) \tag{2.1.3-1}$$

where:

$\Theta(s)$ = actual angular position of the elbow link, (rad)

N(s) = measurement noise for the elbow sensor, (rad)

$$\dot{\hat{\Theta}}(s) = \frac{s}{\tau_v s + 1}\hat{\Theta}(s) \tag{2.1.3-2}$$

where:

τ_v = time constant for the derived angular velocity, (sec)

A control law is necessary to implement the control system. Modified Proportional-Derivative control or modified PD control is used for this thesis (2.1.3-3). As the name implies, this control law is similar to PD Control. The difference is that the derivative gain acts directly on the derivative of the control parameter, rather than the derivative of the error signal. For sudden changes in the desired input, such as a step input, the derivative of the error signal is instantaneously infinite, theoretically. This undesirable effect is mitigated using modified PD control. The control law (where n is measurement noise) is:

$$K_t I_{REF}(s) = T_{REF}(s) \equiv K_p\left[\Theta^*(s) - \hat{\Theta}(s)\right] - K_d\dot{\hat{\Theta}}(s) \qquad (2.1.3\text{-}3)$$

The outer loop gains can be determined using the Zeigler-Nichols Method. This method is described in many control theory textbooks (Shahian, 1993). The PID tuning procedure was followed which determines K_p by setting K_i and K_d to zero. K_m is the value of K_p gain at which the system just oscillates (closed loop poles on the $j\omega$ axis). The oscillation started at a K_p value of ~20, with an oscillation frequency of ~4 Hz. In order to obtain these results, the motor coil saturation had to be temporarily removed from the model (limits set to `±inf`). The controller gains suggested by Zeigler-Nichols are determined in (2.1.3-4) and (2.1.3-5).

$$K_p = 0.6 \cdot K_m = 0.6 \cdot 20 = 12 \qquad (2.1.3\text{-}4)$$

$$K_d = \frac{K_p \cdot \pi}{4 \cdot \omega_m} = \frac{12 \cdot \pi}{4 \cdot (2 \cdot \pi \cdot 4)} = 0.0375 \qquad (2.1.3\text{-}5)$$

For implementation, $K_p = 12$ and $K_d = 0.03$. The actual values of the closed loop natural frequency and damping can be determined to be 19.0 rad/s and 0.308 respectively. The actual system damping will be considerably larger due to the non-linear friction term that was ignored temporarily. Closed loop simulations, which include this non-linear term, will provide a better estimate of system performance. Integral action may be necessary to obtain good tracking

performance especially in the presence of a disturbance that is asymmetrical about zero. This integral action will be held off from implementation unless needed. Some steady state error will likely be encountered.

2.2 Disturbance Observer Design

The advantage of the DOB over classical control algorithms alone is smaller steady-state error in the presence of external disturbances. The disadvantage of the DOB is the potential to inadvertently create an unstable control system. Since the DOB control signal and the outer loop control signals are additive, there exists a possibility of saturating the control actuators. Such elements as D.C. motors are torque limited due to voltage saturation of the motor power electronics. If the DOB controller and the rest of the control system components are being designed and implemented together, the selection of saturation-prone components can be made with a greatly reduced risk. However, if the DOB is being retrofitted into an existing system (as in this thesis) great care must be taken not to create an unstable situation by saturating the actuators.

2.2.1 Introduction to the Disturbance Observer

A block diagram of the DOB is shown in Figure 2.4. $G_p(s)$ represents the actual plant dynamics. $G_n(s)$ is a model of nominal plant dynamics and Q(s) is a specially designed filter, which together form the DOB. Their purpose is to estimate the disturbance based on the incoming signal. The signals are labeled as follows: T_{REF} is the reference command from the outer control loop, T_d is the disturbance, Θ is the actual angular position of the elbow, N is measurement noise, $\tilde{T}_d$ is the disturbance estimate before Q(s), and $\hat{T}_d$ is the disturbance estimate after Q(s).

The DOB approach takes the incoming control signal and the feedback measurement and mathematically derives an estimate of the disturbance signal ($\hat{T}_d$). The algorithm is based on a nominal model, $G_n(s)$. The nominal model predicts the value of the controlled parameter for a given command; the feedback signal contains the information of the actual value of the controlled parameter. Some portion of the feedback signal measurement is influenced by the disturbance (which may be and is generally from multiple sources). The DOB will take these two pieces of information and use them to create an estimate of the disturbance. The derivation of the DOB algorithm follows in the next section. The design issue is choosing or designing Q(s) to provide a good balance between disturbance rejection vs. stability, robustness, and noise sensitivity. In addition, the detail

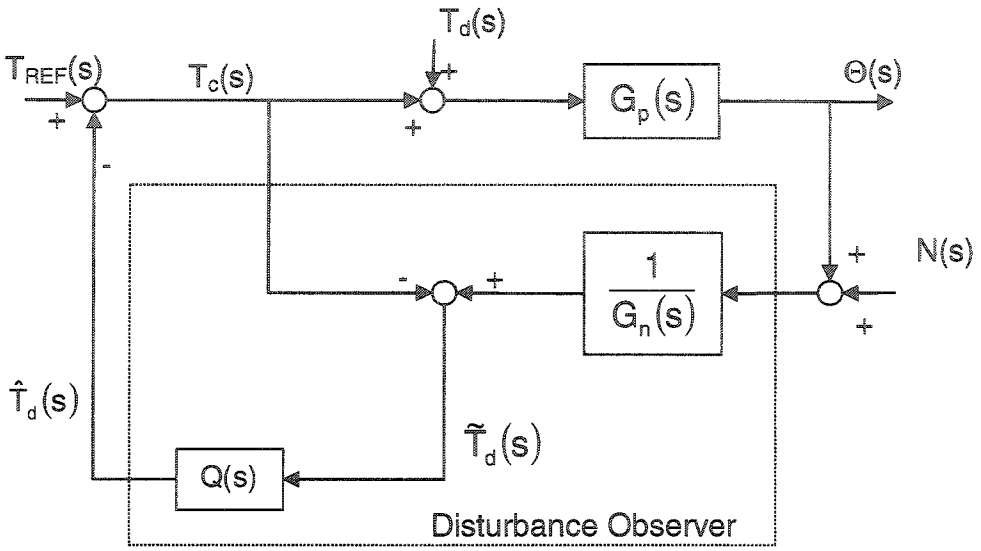

Figure 2.4: The Disturbance Observer Block Diagram

The Disturbance Observer takes the control signal and feedback signals as its inputs and mathematically derives an estimate of the imbedded disturbance signal ($\hat{T}_d$) based on a reference or nominal model (G_n). The disturbance estimate can then be used as a feed-forward signal to cancel the actual disturbance as it enters the plant, (G_p)

contained in the nominal model is equally critical since its contribution must be accurate as well as not overly burdensome to the Digital Signal Processor (DSP), if throughput is low.

It should be noted that although the DOB representation of Figure 2.4 is a convenient way of illustrating the role of Q(s), the implementation of Q(s) can not be accomplished as shown. The inverse of the nominal model transfer function will typically yield a numerator with higher order than its denominator. Since this representation is physically unrealizable, it must be corrected with block diagram

algebra by bringing Q(s) ahead of the summing junction. Therefore, realization of the DOB requires that the order of Q(s) be greater than or equal to that of $G_n(s)$, since the order of the numerator of $Q \cdot G_n^{-1}$ must be greater than or equal to that of the denominator. In addition, care must be taken to insure that the units of parameters entering and leaving the summing junctions have consistent units.

As mentioned previously, this thesis will compare two key types of DOB algorithms. The derivation of the DOB algorithm below applies equally well to either of the two types. Not until specific designs for Q(s) and $G_n(s)$ are discussed will there be any need to distinguish between the types. The equations below can be either vector or scalar equations depending on the application. For example, if the DOB is to be applied to a robotic manipulator with multiple controlled linkages then the equations below would be best written in vector form. In this thesis the DOB will be applied to a SCARA robot in which only the elbow link is utilized. Therefore the equations below will be in scalar form and derived for the elbow link only.

2.2.2 "Real-System" Model

The derivation of equations begins with an expression of the robotic equation in Joint Space (2.2.2-1), (Craig, 1989, chap. 6). This equation is referred to as the "Real-System" model in this thesis.

$$T_C(s) - T_{EXT}(s) = J \cdot \ddot{\Theta}(s) + C\left(\Theta(s), \dot{\Theta}(s)\right) + F\left(\Theta(s), \dot{\Theta}(s)\right) \qquad (2.2.2\text{-}1)$$

where:

J_{EFF} = Elbow effective inertia (constant), oz_f-in-s^2

$\ddot{\Theta}(s)$ = Elbow Angular acceleration, rad/sec^2

$\dot{\Theta}(s)$ = Elbow Angular velocity, rad/sec

$\Theta(s)$ = Elbow Angular position, rad

T_C = Control torque, oz_f-in

T_{EXT} = External disturbance torque, oz_f-in

$C\left(\Theta, \dot{\Theta}\right)$ = Coriolis and centrifugal torque terms, oz_f-in

$F\left(\Theta, \dot{\Theta}\right)$ = Friction torque terms, oz_f-in

The external torque, T_{EXT}, includes torque created by sensor noise, additional mass retrieved by the manipulator, and external torque on the manipulator.

2.2.3 The Nominal Model

At this juncture it is important to note that the key difference between the two types of DOB algorithms presented in this thesis lie in the definition of the nominal dynamics used in the nominal model. The Kaneko DOB approach uses a very simple nominal dynamics model (as shown in "Case 1"), $G_n(s)$, which includes only

inertial dynamics (2.2.3-1) (Kaneko, 1994). The other Q(s) Filter DOB algorithms (as shown in "Case 2"), all use a slightly more detailed nominal dynamics model which includes a viscous damping term (2.2.3-2). The only angular position available to input to the nominal model is the measured parameter, $\hat{\Theta}(s)$ (2.1.3-1). Let the Nominal Model for Case 1 and 2 in transfer function form be as shown in (2.2.3-1) and (2.2.3-2) respectively.

Case 1: (for Kaneko DOB)

$$\text{Let:} \qquad G_{n1}(s) = \frac{\hat{\Theta}(s)}{T(s)} = \frac{1}{J_n s^2} \tag{2.2.3-1}$$

Case 2: (for Q(s) Filter DOB)

$$\text{Let:} \qquad G_{n2}(s) = \frac{\hat{\Theta}(s)}{T(s)} = \frac{1}{J_n s^2 + B_n s} \tag{2.2.3-2}$$

where:

J_n = Nominal inertia (constant),

B_n = Nominal viscous damping (constant)

2.2.4 New "Real System" Model

The "Real System" Model can be expressed in terms of the Nominal Model. The torque defined by the Nominal Model is added to both sides of the Real System Model (2.2.2-1):

Case 1:

$$J_n s^2 \hat{\Theta} = (T_C - T_{EXT}) + (J_n s^2 \hat{\Theta} - J s^2 \Theta) - C(\Theta, \dot{\Theta}) - F(\Theta, \dot{\Theta}) \qquad (2.2.4\text{-}1)$$

$$J_n s^2 \hat{\Theta} = T_C + T_d \qquad (2.2.4\text{-}1a)$$

Case 2:

$$J_n s^2 \hat{\Theta} + B_n s \hat{\Theta} = (T_C - T_{EXT}) + (J_n s^2 \hat{\Theta} - J s^2 \Theta) + B_n s \hat{\Theta} - C(\Theta, \dot{\Theta}) - F(\Theta, \dot{\Theta}) \qquad (2.2.4\text{-}2)$$

$$J_n s^2 \hat{\Theta} + B_n s \hat{\Theta} = T_C + T_d \qquad (2.2.4\text{-}2a)$$

T_d in (2.2.4-1a) and (2.2.4-1b) can be derived as:

Case 1:

$$T_d = \left(J_n s^2 \hat{\Theta} - J s^2 \Theta\right) + B_n s \hat{\Theta} - C\left(\Theta, \dot{\Theta}\right) - F\left(\Theta, \dot{\Theta}\right) - T_{EXT} \qquad (2.2.4\text{-}3)$$

Case 2:

$$T_d = \left(J_n s^2 \hat{\Theta} - J s^2 \Theta\right) - C\left(\Theta, \dot{\Theta}\right) - F\left(\Theta, \dot{\Theta}\right) - T_{EXT}$$

T_d contains all of the differences between the nominal inertia and the actual inertia of the real system, the friction, and any difference between the nominal external torque and the real external torque. This disturbance torque is known to deteriorate the performance of a SCARA control system. The goal of the disturbance observer is to develop an estimate of this disturbance torque and use it to cancel with (2.2.4-3).

2.2.5 Estimated Disturbance Torque

Using the DOB, an estimate of the disturbance torque, $\hat{T}_d$, is created as a cancellation signal to create a new control torque. This disturbance estimate is added to the reference torque from the PD controller, as shown in (2.2.5-1), to cancel (or suppress) the actual disturbance torque.

$$T_C \equiv T_{REF} - \hat{T}_d \qquad (2.2.5\text{-}1)$$

where:

T_C = New control torque (modified PD + DOB)

T_{REF} = Control torque (modified PD)

$\hat{T}_d$ = Disturbance Observer estimated disturbance torque

An estimate of the disturbance torque can be derived from the New Real System Models (2.2.4-1a and 2.2.4-2a), using the general nomenclature for the nominal model (G_n). T_C can be substituted in (2.2.4-3) for the original control torque, T_{REF} as shown in (2.2.5-2).

$$\tilde{T}_d = \frac{1}{G_n}\hat{\Theta} - T_C \qquad (2.2.5\text{-}2)$$

where:

$\tilde{T}_d$ = Derived disturbance torque estimate

T_C = New control torque (PD and DOB)

$\hat{\Theta}$ = Measured angular position

G_n = Nominal Model, G_{n1} or G_{n2}

The measured parameter generally available is angular position via encoder measurement. However, the double differentiation of the angular position in

(2.2.5-2) due to the inverse of G_{n1} or G_{n2} is not desirable since differentiating the noise, which accompanies position measurements, creates higher amplitude noise leading to control system instability. The practical implementation of the DOB includes a filter, Q(s). This approach has the added advantage of being able to tailor the behavior of Q(s) to match the unique needs of the plant dynamics on a case-by-case basis. It is also important to note that the cutoff frequency of Q(s) can be adjusted independently of the PD controller gains. So although the added complexity of the DOB controller should not be taken lightly, the tuning of the outer loop PD gains will still utilize techniques developed in Classical Control, while the tuning of the DOB is handled separately. The resulting estimated disturbance torque using Q(s) is then:

$$\hat{T}_d = Q(s) \cdot \tilde{T} \qquad (2.2.5\text{-}3)$$

For this thesis, the nominal dynamics model shown in (2.2.3-1) along with a second order Q(s) filter represents the Kaneko DOB algorithm. The nominal dynamics model shown in (2.2.3-2) along with a second, third or fourth order Q(s) filter, will be referred to as the Q(s) Filter DOB algorithm. There is no rule to limit the paring of the Kaneko $G_n(s)$ in (2.2.3-1) with Q(s) filters of order higher than two. Close examination of these two nominal models reveals that the Q(s) Filter DOB algorithms require knowledge of the nominal system damping, B_n. The beauty of the Kaneko nominal model is that it does not require any knowledge of the system

damping to implement the DOB. Moreover, the Kaneko DOB treats all un-modeled torques as disturbances.

2.3 Characteristics of the Disturbance Observer

The selection of the low pass filter type for Q(s) over other filter types will become apparent when the properties of the observer are considered. This is more easily accomplished using an equivalent block diagram to Figure 2.4. The block diagram in Figure 2.5 results with modest block diagram reduction effort. Three useful equations can be derived by alternately setting two of the three external inputs (command, disturbance, and noise) to zero and redrawing the figure. These equations are key to understanding the behavior of the DOB.

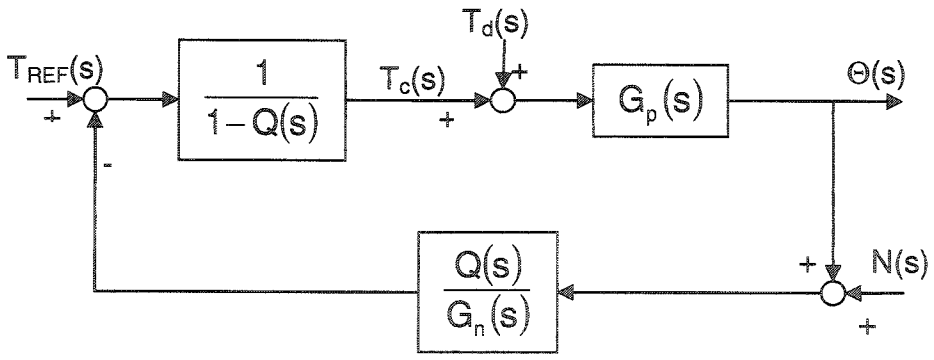

Figure 2.5: Equivalent Disturbance Observer Block Diagram

The Disturbance Observer is this more convenient equivalent form from which to derive various transfer functions describing system behavior relative to the output parameter.

$$G_{T\Theta}(s) \equiv \frac{\Theta(s)}{T_{REF}(s)} = \frac{G_p(s) \cdot G_n(s)}{G_n(s) + Q(s)[G_p(s) - G_n(s)]} \tag{2.3-1}$$

$$G_{D\Theta}(s) \equiv \frac{\Theta(s)}{T_d(s)} = \frac{G_p(s) \cdot G_n(s)[1 - Q(s)]}{G_n(s) + Q(s)[G_p(s) - G_n(s)]} \tag{2.3-2}$$

$$G_{N\Theta}(s) \equiv \frac{\Theta(s)}{N(s)} = \frac{G_p(s) \cdot Q(s)}{G_n(s) + Q(s)[G_p(s) - G_n(s)]} \tag{2.3-3}$$

Certain useful characteristics can be determined from (2.3-1) through (2.3-3) by observing their behavior when the Q(s) approaches 1 (no filtering effect) and also when Q(s) approaches 0 (perfect filtering effect). A study of the above equations reveals that:

As Q(s) approaches 1:

$G_{T\Theta}(s) \rightarrow G_n(s)$ (from 2.3-1)

$G_{D\Theta}(s) \rightarrow 0$ (from 2.3-2)

As Q(s) approaches 0:

$G_{N\Theta}(s) \rightarrow 0$ (from 2.3-3)

Q(s) should be designed such that it suppresses low frequency disturbances, and filters out high frequency measurement noise. This constraint will be accomplished if Q(s) approaches one for low frequencies (2.3-1) and Q(s) approaches zero for high frequencies (2.3-3). Since the low pass filter has these desired characteristics at low and high frequency, it is the best choice for the Q(s) filter. In addition, review of (2.3-2) shows that the disturbance is driven by the parameter [1 – Q(s)], while measurement noise attenuation is driven by Q(s) (2.3-3). A study of the Bode plots of these two parameters will demonstrate how best to make use of these characteristics in the design of the DOB.

2.3.1 Q(s) Filters

The selection of the Q(s) filter is one of the most important steps in the design of the DOB servo system, (Dote, 1998, p.58). The design of new Q(s) filters is well beyond the scope of this thesis, however a comparison of the performance of several previously designed Q(s) filters will be shown. Four Q(s) filters were selected on the basis of studying the effect of varying the order has on performance. The selection consists of one second order, two third order, and one fourth order filter as shown by equations (2.3.1-1) through (2.3.1-4).

The Kaneko disturbance observer utilized a second order filter of the form shown in (2.3.1-1). The first digit of the subscript "20" as in $Q_{20}(s)$ represents the order of the denominator, while the second digit refers to the order of the numerator.

$$Q_{20}(s) = \frac{g_2}{s^2 + g_1 s + g_2} \tag{2.3.1-1}$$

where:

g_1 = cutoff frequency tuning parameter, ($g_1 = 2\zeta\omega_n$)

g_2 = cutoff frequency tuning parameter, ($g_2 = \omega_n^2$)

ζ = damping ratio ($\zeta = 1.0$)

ω_c = cutoff frequency, rad/s ($\omega_c = 2\pi f_c$ $^{rad}/_{s}$)

ω_n = filter natural frequency, rad/s ($\omega_n = \alpha_n \omega_c$)
for:
$\zeta = 1.0$: $\alpha_c = 1.55$
$\zeta = 0.707$: $\alpha_c = 1$

f_c = cutoff frequency, Hz

The third and fourth order filters are:

$$Q_{31}(s) = \frac{3\tau s + 1}{\tau^3 s^3 + 3\tau^2 s^2 + 3\tau s + 1} \tag{2.3.1-2}$$

$$Q_{31D}(s) = \frac{3 \cdot 3\tau s + 1}{3^3\tau^3 s^3 + 3 \cdot 3^2\tau^2 s^2 + 3 \cdot 3\tau s + 1} \qquad (2.3.1\text{-}3)$$

$$Q_{42D}(s) = \frac{16 \cdot 7.7\tau^2 s^2 + 4 \cdot 5.1\tau s + 1}{256\tau^4 s^4 + 64 \cdot 4.7\tau^3 s^3 + 16 \cdot 7.7\tau^2 s^2 + 4 \cdot 5.1\tau s + 1} \qquad (2.3.1\text{-}4)$$

The third order Q(s) filter (2.3.1-2) is the well known $Q_{31}(s)$ filter, (Umeno and Hori,1991) and (Kempf and Kobayashi, 1999). The other third order filter, $Q_{31D}(s)$, (2.3.1-3) and the fourth order, $Q_{42D}(s)$ (2.3.1-4) filter designs are documented by (Dote, 1998, p58), and (Umeno and Hori,1991). $Q_{31D}(s)$ and $Q_{42D}(s)$ are based on Butterworth filter approach according to (2.3.1-5). The subscript nomenclature convention described for (2.3.1-1) is continued for these higher order Q(s) filters, where the "D" in $Q_{31D}(s)$ and $Q_{42D}(s)$ stands for Dote to distinguish from the $Q_{31}(s)$ design. It should be noted that $Q_{31D}(s)$ can be obtained from $Q_{31}(s)$ by substituting 3τ for τ in (2.3.1-2). Therefore, since these two filters are identical, only the $Q_{20}(s)$, $Q_{31D}(s)$, and $Q_{42D}(s)$ will be presented.

$$Q(s) = \frac{1 + \sum_{k=1}^{N-2} a_k (s\tau N)^k}{1 + \sum_{k=1}^{N} a_k (s\tau N)^k} \qquad (2.3.1\text{-}5)$$

where:

$N \geq 3$ = order of Q(s)

τ = cutoff frequency tuning parameter, $\tau = k_c / \omega_c$
for:
$Q_{31}(s)$: $k_c = 1.64$
$Q_{31D}(s)$: $k_c = 0.547$
$Q_{42}(s)$: $k_c = 0.683$

k_c is found by varying τ parametrically in each of the Q(s) filters as a function of the -3db frequency.

ω_c = cutoff frequency, (rad/sec)

s = Laplace differentiator

a_k = coefficients of parameters Q(s), Table 2.1

Table 2.1: Coefficients of Parameters $Q_{31D}(s)$ and $Q_{42D}(s)$

N	a_1	a_2	a_3	a_4
3	3	3	1	-
4	5.1	7.7	4.7	1

2.3.2 Q(s) Filter Cutoff Frequency

The cutoff frequency for the filters shown in this thesis is defined as the frequency of a sinusoidal input where the output amplitude is attenuated by -3db. The desired cutoff frequency is expressed in rad/s as $\omega_c = 2\pi f_c$, where f_c is the

cutoff frequency in Hertz. The cutoff frequency for the second order Q(s) filter is specified by its tuning parameters, $g_1 = 2\zeta\omega_n$ and $g_2 = \omega_n^2$, where ω_n is the natural frequency which is related to the cutoff frequency by $\omega_n = \alpha_n\omega_c$ and α_n is a function of the damping ratio, ζ; for $\zeta = 1$ as in this thesis, $\alpha_n = 1.55$ (Raven, 1995, p. 537).

A typical Bode plot for the second order $Q_{20}(s)$ in (2.3.1-1) with a cutoff frequency of 40 Hz is shown in Figure 2.6, where the Bode set in red is for Q(s), while the set in blue is for [1 – Q(s)].

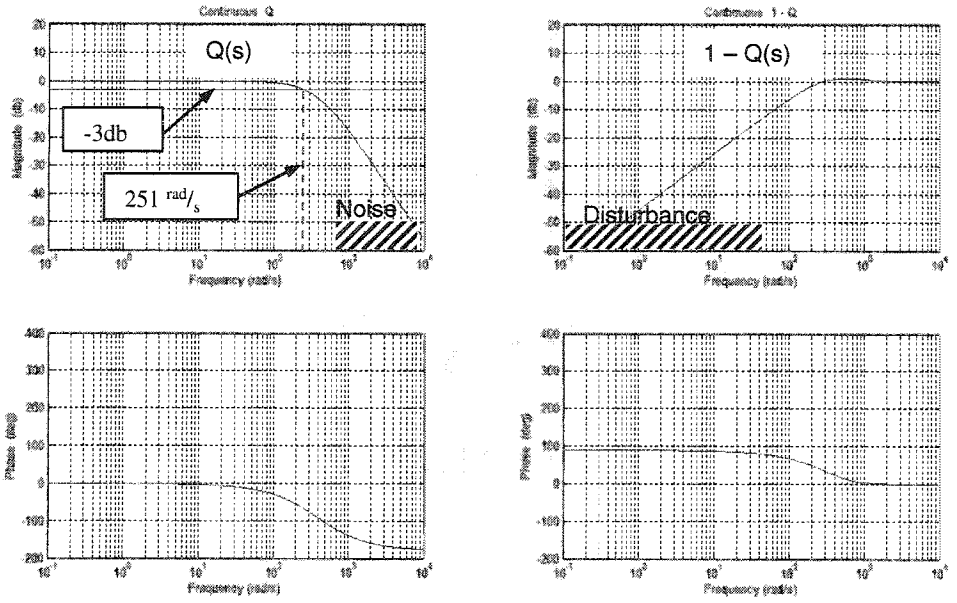

Figure 2.6: Frequency Response of $Q_{20}(s)$ Filter (f_c = 40 Hz)

Bode plot of the Disturbance Observer $Q_{20}(s)$ filter shows the disturbance and noise attenuation properties of the low pass filter.

The result is high attenuation at low frequencies for [1 – Q(s)] which drives the disturbance. The plot also shows high attenuation at high frequencies for Q(s), which drives the measurement noise. Therefore Q(s) is tuned by adjusting the cutoff frequency (ω_c) to achieve a good tradeoff between disturbance and noise attenuation. It is possible that good performance is unrealizable or that too much compromise must be endured for a given test setup. In this event it may be necessary to consider a higher order Q(s). Since frequently phase margin tends to decrease as filter order increases, a good general design practice would select the lowest order filter which results in acceptable performance. It will be seen later that the cutoff frequency cannot be set arbitrarily high to get as much bandwidth as possible. Stability considerations place an upper limit on the cutoff frequency setting of Q(s).

The cutoff frequency tuning parameters for the third and fourth order filters is determined by $\tau = k_c/\omega_c$, where k_c is given for each of these filters in 2.3.1-5. The k_c values are found by varying τ parametrically in each of the Q(s) filters as a function of the -3db frequency (cutoff frequency). The relationship between ω_c and τ is determined from the parametric data.

A comparison can be performed (in continuous time) of the effect of Q(s) filter order on DOB performance. Figure 2.7 shows two Bode plot sets (Q and 1 – Q) for three Q(s) filters: the second order $Q_{20}(s)$ (blue), third order Dote $Q_{31D}(s)$ (black) and the fourth order Dote $Q_{42D}(s)$ (red). The cutoff frequency for all of the Q(s) filters (in Figure 2.7) is 100 hz (628 rad/sec) for comparison purposes only (as

indicated by the intersection of the Q(s) magnitude curves at -3db). The Q(s) filter plots are nearly overlays of one another. From the 1 – Q(s) Bode magnitude plots, it would appear that as order increases, so does the bandwidth for disturbance rejection. However, the phase behavior must also be taken into account. A reverse trend is observed for the phase behavior: increasing phase lead as order increases. The ultimate selection of the Q(s) order for a given application must take into account the affect of Q(s) on the performance of the overall system.

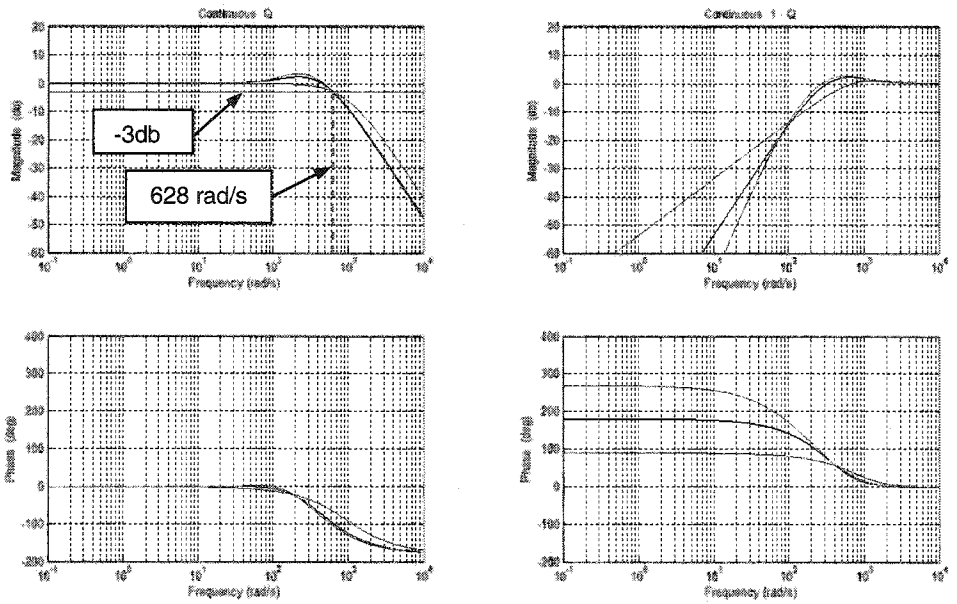

Figure 2.7: Comparison of the Order of Q(s) in Continuous Time, $Q_{20}(s)$ (blue), $Q_{31D}(s)$ (black), $Q_{42D}(s)$ (red)

Any unwanted signal entering either the forward loop or the feedback loop is "disturbing." So then a definition is needed to describe the use of the terms disturbance and measurement noise: disturbance is defined as a signal of large amplitude and low frequency, while measurement noise is defined as a signal of small amplitude and high frequency which enters the system via the feedback sensor. Large amplitude vs. small amplitude and low frequency vs. high frequency must be defined on a case-by-case basis. In general, disturbances can cause large scale, undesirable plant motions, which at the extreme can result in structural damage. Noise on the other hand can mask command and control signals limiting accuracy at the steady-state level. As an anecdotal example: wind can be considered a disturbance to a ground-based dish antenna. Gusts can be of such a magnitude to either create a loss of pointing accuracy sufficient to loose the signal track, or at the extreme, damage the antenna structure, requiring shutdown to prevent such loss. Measurement noise may be sufficiently large as to create steady-state position or tracking errors large enough to prevent the antenna from locking-on to the incoming signal. Either or both of these can be showstoppers in a given design. The goal of the DOB is to suppress the disturbance signal and attenuate noise signals enough to extend or permit the use of the equipment.

2.3.3 Open and Closed Loop Transfer Functions and Stability

Open loop analysis allows the determination of the open loop gain and phase margins. In a system so rich in non-linearity, especially motor saturation, analytical results are approximate at best. As will be discussed later, the maximum voltage allowed by the DSP board in this implementation is 10 volts. For a bit of safety margin, the overwhelming majority of runs were made at 5 volts. Therefore, the actuator is expected to be saturated for some portion of a typical step input command.

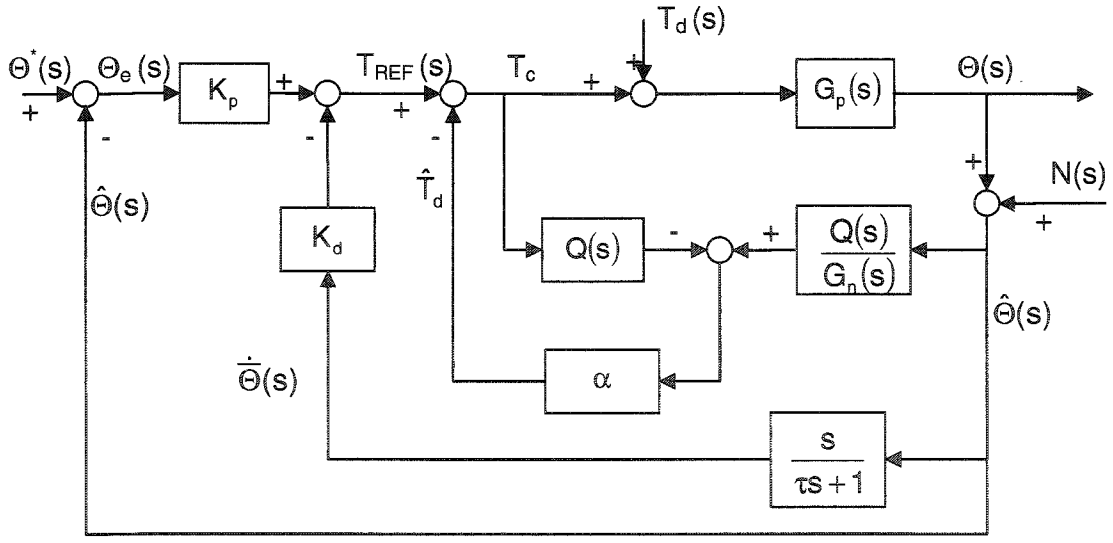

Figure 2.8: Closed Loop Model

Open and closed loop transfer functions are derived from the closed loop model with the disturbance observer and the outer modified PD control loop. A new parameter, α, is introduced to prevent the estimated disturbance signal from causing the control system to become unstable.

In addition to the closed loop model (Figure 2.8), a new gain parameter, alpha (α), is introduced. The purpose of α is to prevent the combination of the outer loop and DOB control signals from creating an unstable system.

The open loop transfer function from the error signal to the angular position output is given by (2.3.4-1):

$$\frac{\Theta(s)}{\Theta_e(s)} = \frac{G_p \cdot G_{c1}}{1-\alpha \cdot Q + \alpha \cdot Q \cdot G_n^{-1} \cdot G_p + G_p \cdot G_{c2}} \qquad (2.3.4\text{-}1)$$

where:

$$G_{c1} = K_p$$

$$G_{c2}(s) = K_d \cdot \frac{s}{\tau s + 1}$$

The closed loop transfer function from $\Theta^*(s)$ to $\Theta(s)$ can also be derived from Figure 2.8 and is given by (2.3.4-2).

$$\frac{\Theta(s)}{\Theta^*(s)} = \frac{G_p \cdot G_{c1}}{1-\alpha \cdot Q + \alpha \cdot Q \cdot G_n^{-1} \cdot G_p + G_p \cdot (G_{c1} + G_{c2})} \qquad (2.3.4\text{-}2)$$

The real significance of α can be realized by observing at low frequencies as Q(s) approaches one, the $(1-\alpha Q)^{-1}$ term becomes infinite for $\alpha=1$ (as shown

in Figure 2.9). The plant will be receiving a control torque that is at the saturation level of the power amplifier.

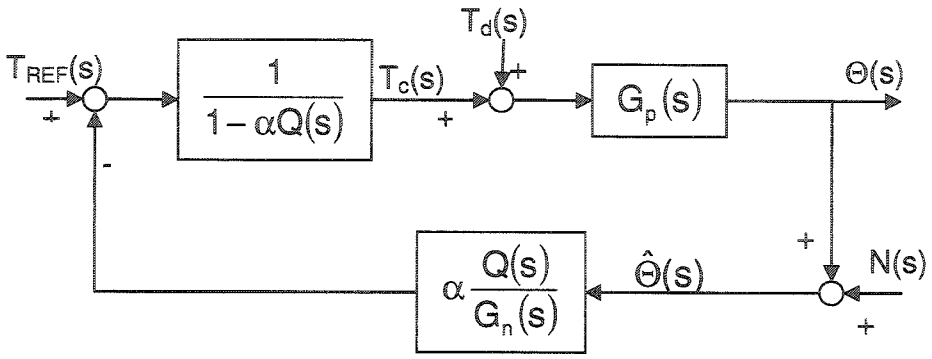

Figure 2.9: Disturbance Observer Loop Showing α Effect

Therefore, for stability of the closed loop system, alpha should be less than one, but as close to one as possible to maximize the disturbance rejection potential from the DOB. For the continuous system, gain and phase margins are independent of α (provided $G_n = G_p$).

2.3.4 Open Loop Results

Gain Margin (GM) and Phase Margin (PM) can be approximated using open loop analysis. A Matlab script (dob_design4c.m, Appendix B.1) was written to create the open loop Bode plot in continuous space, for (2.3.4-1). The GM is large and the PM is 34.1° at 17.465 rad/s (2.78 Hz), (as shown in Figure 2.10). This

model must be viewed as an approximation since this approach lacks the non-linear friction and saturation elements. $Q_{31}(s)$ is used in this analysis.

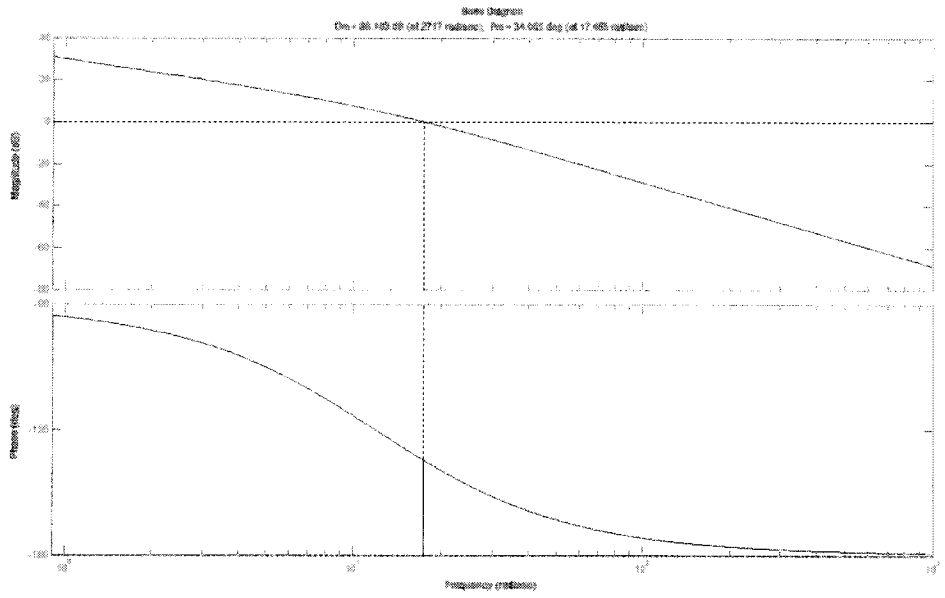

Figure 2.10: Open Loop Gain and Phase Margin – Continuous Time GM = 86 db and PM = 34° at 17.465 rad/s (2.78 Hz)

Gain and phase margins give an indication of how far a system is from critical stability points. In this case the gain is very large; the phase margin is defined at the gain crossover frequency as shown and indicates how far the system is from the phase limit of -180°.

2.4 Modeling and Simulation

The design approach for the DOB control system used for this thesis will be to obtain good performance using a continuous time model and the many well established design techniques available. Once this has been achieved, the

controller will be converted to discrete time. Well-established techniques and analytical tools are also available in Matlab for obtaining discrete time equivalents of system components (transfer functions) from continuous time counterparts.

2.4.1 Continuous Closed Loop Simulation Model

The Matlab / Simulink model for the system is shown in Figure 2.11. This model is based on the plant model (2.1.2-4), the outer loop modified PD controller (2.1.3-3) and the DOB controller (2.2.5-3). Since the control parameter is control current, I_c, the disturbance torque, T_d, and the estimated disturbance torque, $\hat{T}_d$, are converted to current parameters by dividing each by the motor torque constant, K_t. This extra step reduced the number of blocks in the Simulink model slightly. K_t is included in the plant model, G_p and the nominal model.

Two external inputs are included in the model: the disturbance in the forward loop labeled I_d is the large amplitude low frequency disturbance; the block in the feedback loop labeled measurement noise is the small amplitude high frequency noise disturbance (not used in this thesis). The disturbance I_d is modeled as an asymmetrical rectangular disturbance pulse train with a period of one second, a 50% duty cycle, and an amplitude range from 0 to 0.5 A. This simulated disturbance is imported into the system via software, since natural disturbances in the lab were far too subtle to be impressive.

Measurement noise is not addressed directly, either by detailed analytical study or experimental testing. The optical encoders used in the SJSU SCARA lab test setup are opto-isolated devices, and do not present much of a measurement noise problem. Low noise feedback measurement devices should be high priority for any DOB application, due to the double differentiation in the $G_n(s)^{-1}$ block in (2.2.3-1) and in Figure 2.4. Although the DOB algorithm is a clever means of implementing this block (shown later in Figure 2.7), the double differentiation is still being performed which could cause serious problems if low quality sensors are chosen.

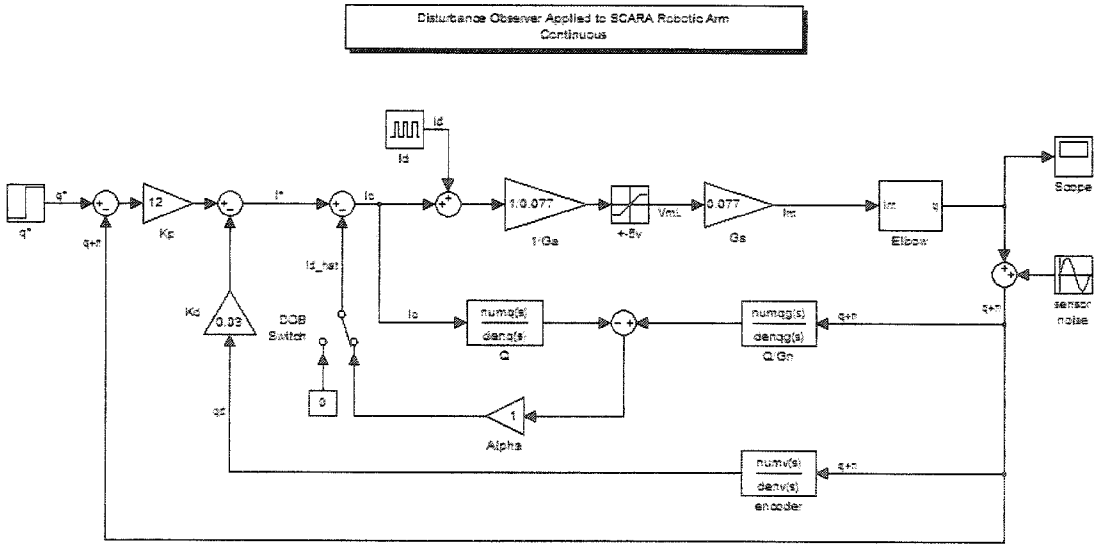

Figure 2.11: Continuous Matlab / Simulink Model for the SJSU SCARA

This diagram shows the actual code used to simulate the SJSU SCARA system in continuous space. The DOB transfer functions are input prior to execution and so this one model is sufficient to run any and all of the DOB designs presented earlier.

There are three blocks in the forward loop (Figure 2.11) whose purpose is to simulate the saturation of the D.C. motor. To do this, the control signal in amps must be divided by the amplifier gain G_a for the elbow link to obtain the motor coil voltage. This voltage was limited to ±5.0 volts by the saturation block. However, on occasion this limit was changed to ±10.0 volts (which is the maximum that the dSpace© would allow).

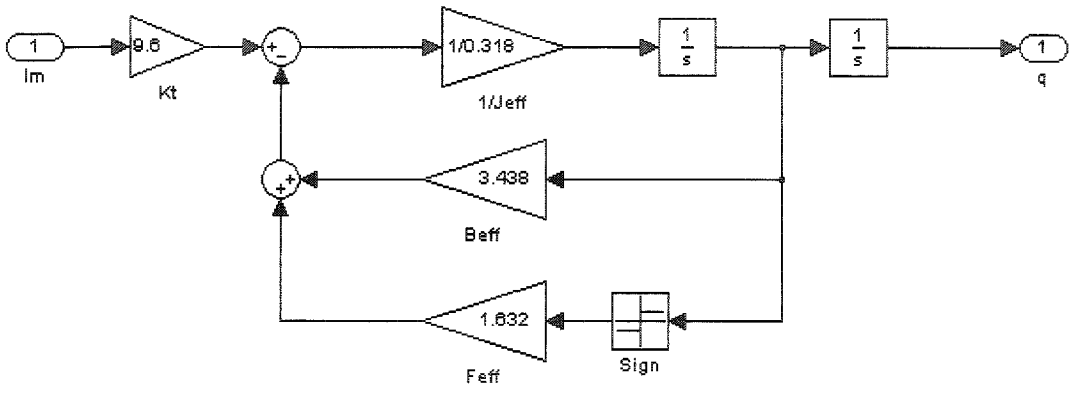

Figure 2.12: Continuous Matlab / Simulink Plant Model for SJSU SCARA
This diagram shows the actual code used to simulate the SJSU SCARA plant model in continuous space. The model includes the viscous and Coulomb elements measured in the laboratory.

The Simulink block diagram of the plant model, $G_p(s)$ (2.1.2-4) is shown in Figure 2.12. The plant model includes the elbow link angular velocity as shown. This parameter could be used as a feedback input to the modified PD controller. However, since the actual SCARA has no such measurement, it is a bit more realistic to model the link velocity as in the actual hardware setup.

The implementation of the DOB in the Simulink model differs from the theoretical block diagram (Figure 2.4) in that a) Q(s) has been moved forward of the summing junction as shown in Figure 2.13, and b) the torque parameters are converted to current parameters by substituting $T(s) = K_t\, I(s)$ in equations (2.2.3-1) and (2.2.3-2) For the case of the Kaneko DOB, a very simple nominal model is used as shown in (2.4.1-2). The Q(s) block is the implementation of $Q_{20}(s)$, $Q_{31D}(s)$, or $Q_{42D}(s)$.

$$G_{n1}(s) = \frac{\hat{\Theta}(s)}{I(s)} = \frac{K_t}{J_n \cdot s^2} \qquad (2.4.1\text{-}2)$$

Then the $\frac{Q(s)}{G_n(s)}$ block is as shown in (2.4.1-3).

$$\frac{Q(s)}{G_n(s)} = \frac{Q_{20}(s)}{G_n(s)} = \frac{J_n \cdot g_2 \cdot s^2}{K_t\left(s^2 + g_1 s + g_2\right)} \qquad (2.4.1\text{-}3)$$

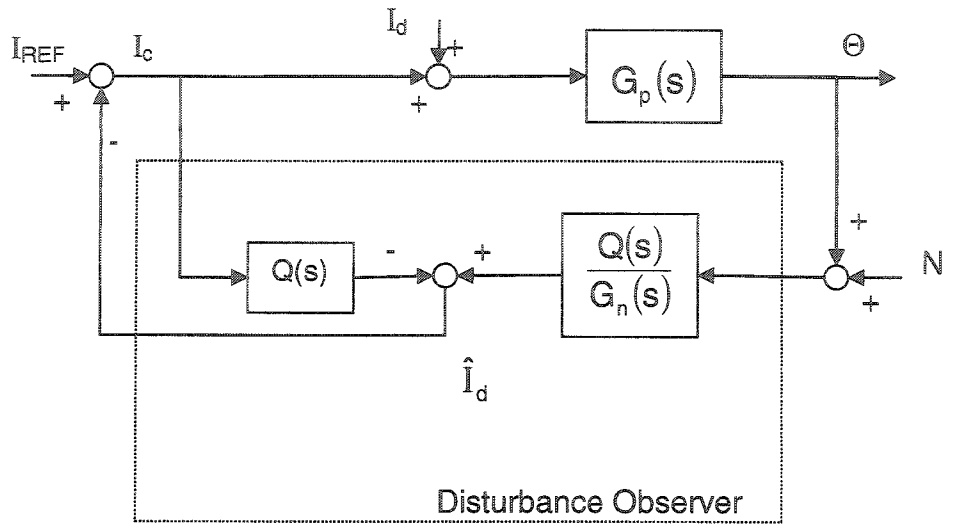

Figure 2.13: Disturbance Observer Model for Simulation

An on-off DOB switch (Figure 2.11) is used in the simulations without the DOB at first to design and test the outer loop gains K_p and K_d, and later to compare system performance with and without the DOB. One key advantage of the DOB is that it allows independent tuning of disturbance rejection characteristics and command following characteristics, (Kempf, 1999). Further, compared to integral action, the DOB allows more flexibility via the selection of order, relative degree, and bandwidth of the Q(s) Filter.

2.4.2 Continuous Closed Loop Simulation Results

Figure 2.14a shows continuous simulation results for the closed loop SCARA system (Figure 2.11) to a 1.0 radian step input with both DOB and disturbance

turned off. The four plots clockwise from the top left represent the angular position feedback in radians, the error signal in radians, the DOB and disturbance in amps (both off), and the angular velocity in radians/second. The settling time of ~1.8 seconds is considerably slower than the goal set earlier at 0.6 seconds; the steady-state error is less than 0.1 degree with minimal overshoot. The settling time results observed in this simulation will not adversely affect the performance of the DOB design. Therefore the outer loop design is acceptable.

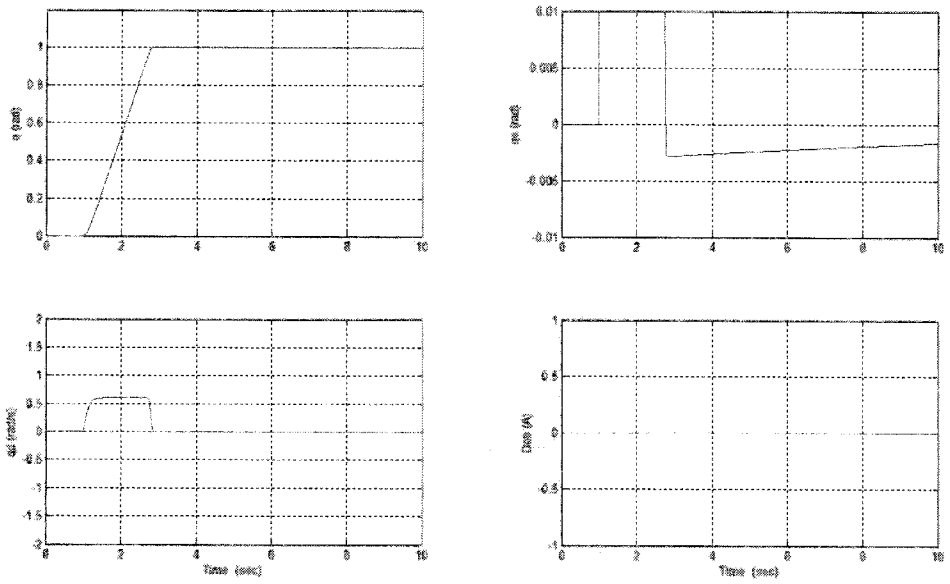

Figure 2.14a: Continuous Closed-Loop Response
DOB Off and Disturbance Off

Figure 2.14b shows simulation results for the same system as above to a 1.0 radian step input with a 0.5 amp disturbance. The DOB is again switched off in

this simulation. The disturbance results in an angular position amplitude in excess of -0.035 rad, or 2.01°. The goal of the DOB algorithm designs is to improve on this result.

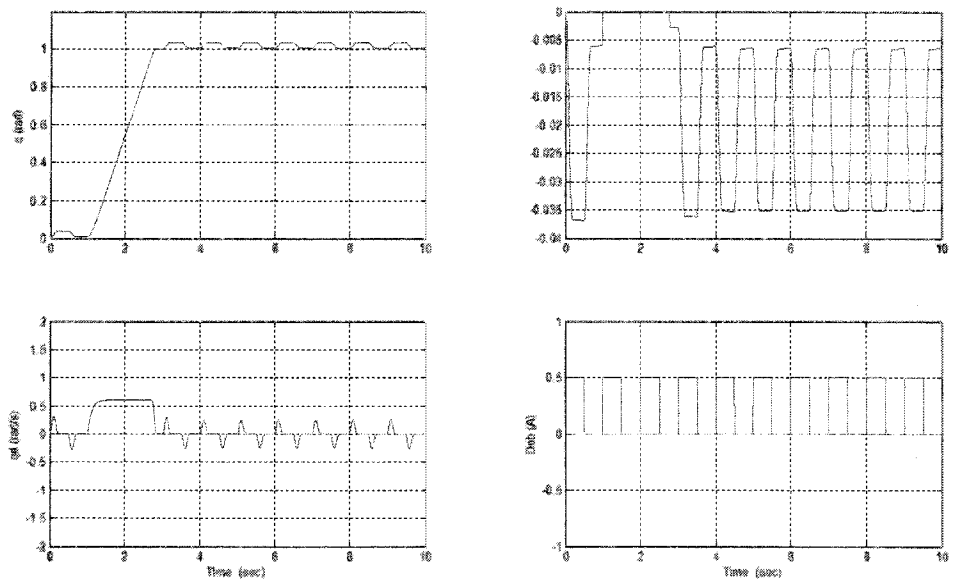

Figure 2.14b: Continuous Closed-Loop Response
DOB Off and Disturbance On

The next challenge will be to invoke the DOB to suppress the disturbance. Figure 2.15a shows the performance of the same system as above with the DOB switched on and $\alpha = 0.90$. The Kaneko DOB algorithm is used in this case. The nominal model used in this algorithm contains no damping and $Q_{20}(s)$ has a cutoff frequency of 57 Hz. The steady-state error has dropped to well under 0.005 rad. with minimal evidence of the disturbance remaining in the feedback signal. The

DOB plot in the compares the actual disturbance (blue) to the estimate by the DOB (green). As shown, the observer locks on and matches the disturbance very well.

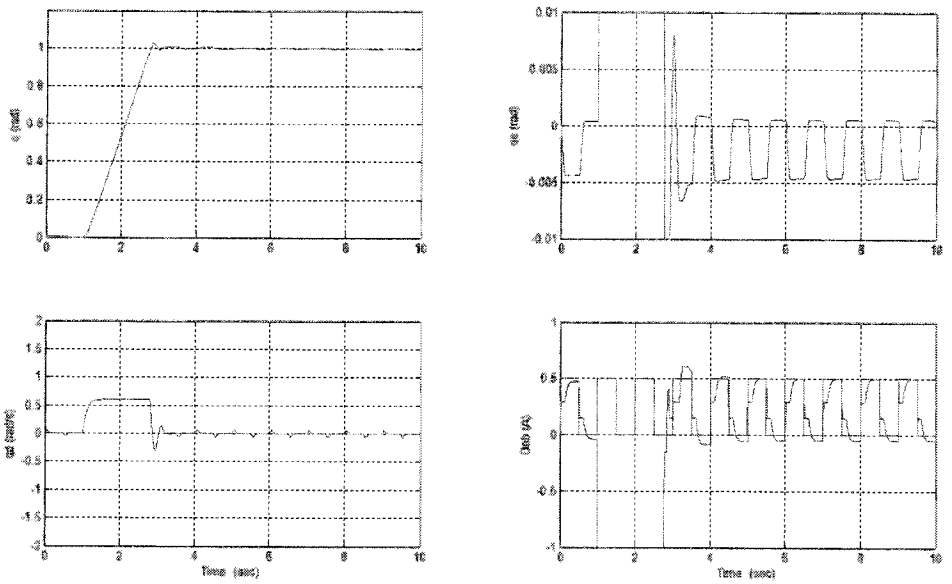

Figure 2.15a: Continuous Closed-Loop Response
DOB and Disturbance On
$Q_{20}(s)$ Filter Algorithm (Kaneko DOB)

It is now important to compare this result to the second order Q(s) Filter DOB algorithm (Figure 2.15b). This algorithm will also use $Q_{20}(s)$ as above, however the nominal model will include a nominal damping term the same as the measured viscous damping value as in (3). The cutoff frequency for this filter is also the same as the previous case (57 Hz). A comparison of the error plots in Figures 2.15a and 2.15b reveals that the response in this case is delayed a bit due the presence of the damping term in the DOB algorithm. However, this DOB has

comparable disturbance matching characteristics as the Kaneko DOB.

Continuous time simulations for the remaining DOB algorithms are in Appendix C.

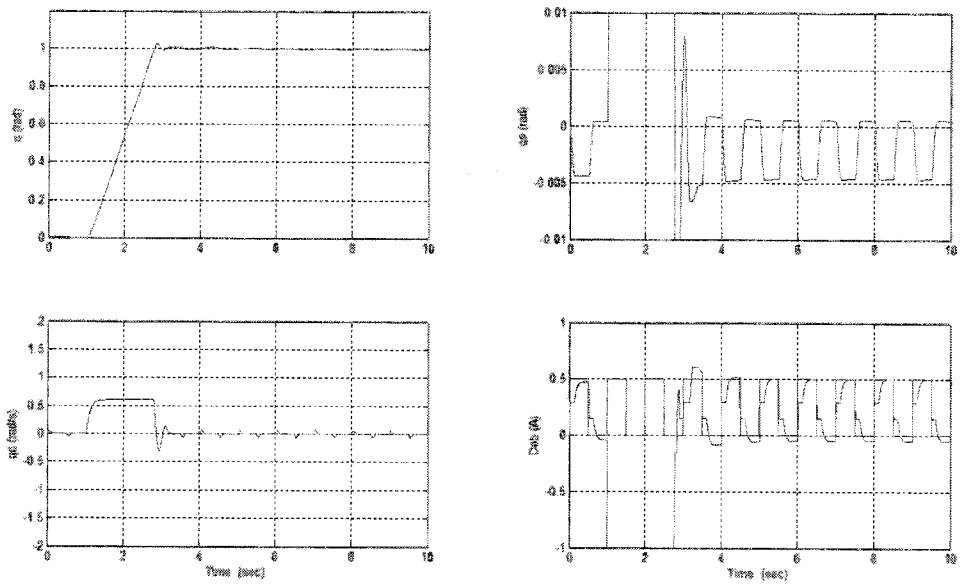

Figure 2.15b: Continuous Closed-Loop Response
DOB and Disturbance On
$Q_{20}(s)$ Filter DOB

2.4.3 Discrete Closed Loop Simulation Model

At this point it is appropriate to convert the model from continuous time to discrete time and rerun the above study. Figure 2.16 shows the SJSU SCARA discrete time version of the preceding model. As can be seen little has changed except to replace continuous transfer function blocks for their discrete equivalents. The plant is inherently continuous time and can therefore be represented as such

even if the controller is in discrete time. For this reason the continuous SCARA dynamics subsystem block can be left unchanged in the discrete model.

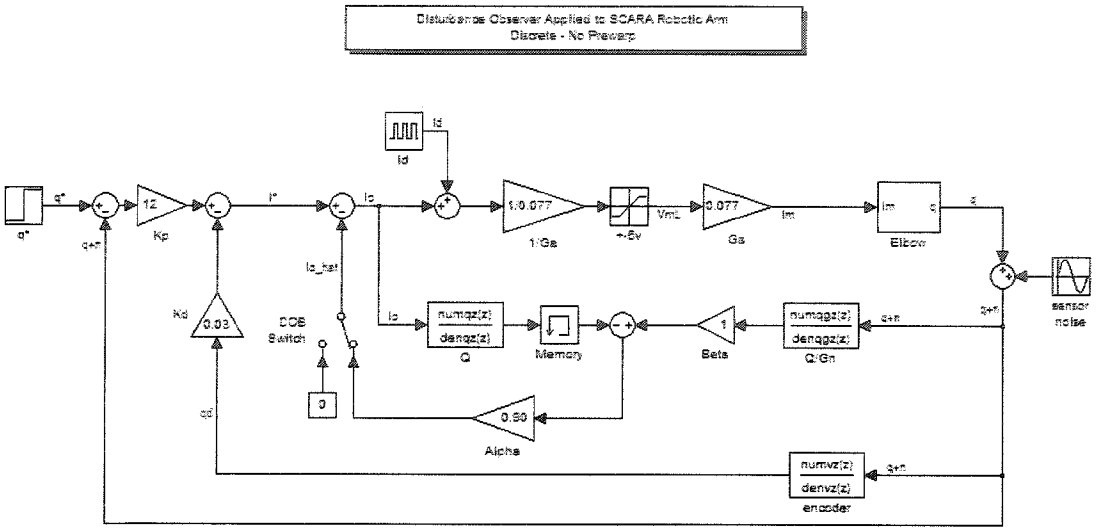

Figure 2.16: Discrete Matlab / Simulink Model for the SJSU SCARA

The selection of the sampling time, T_s, is essential in the discrete model. The closed loop natural frequency was determined earlier to be 19 rad/s. The associated characteristic time is 52 msec. The sample time should be less than the characteristic time. Previous projects using this same SCARA setup set their sampling time at 10msec. Using the encoder resolution to set the sample time resolution seems reasonable, so that the digital controller does not penalize the resolution of the system. In this case the elbow link can traverse ~0.1° in 3msec under a disturbance load of 0.5A. The minimum T_s allowed by the dSpace© 1102 DSP board was determined by experiment to be 3msec. So as will be shown

later, a sample time of 3msec should be adequate based on disturbance rejection criteria.

There is another consideration to be made. The sampling frequency should be set at twice the highest expected control signal frequency to prevent aliasing, (Franklin, 1998, p. 168). For T_s = 3msec the sampling frequency $\omega_s = 2\pi/T_s \approx$ 2094 rad/s. Therefore the highest frequency signal which can be recovered after sampling is the Nyquist frequency, $\omega_N \approx 1047$ rad/s ≈ 167 Hz, which is one-half the sampling frequency. Since no physical process behaves perfectly, a somewhat lower frequency will result. This should not pose any problems since all Q(s) filter cutoff frequencies are well below ω_N. However, if a sampling time of 10msec was chosen, the corresponding Nyquist frequency would be ~50 Hz which is at or below the desired cutoff frequency of some of the Q(s) filters in this thesis. Therefore the minimum sample time of 3msec is adequate for the purpose of this thesis.

Each of the three transfer functions in this model are discretized using Matlab's c2dm function. This function performs the discretization using the zero order hold (ZOH) method by default. This method is appropriate in situations where a zero order hold is present in the continuous loop as is the case for the discretization of the velocity transfer function. However, if no ZOH is present in the loop then the ZOH default c2dm method is inappropriate, as is the case for the Q(s) and the $Q(s)^*G_n(s)^{-1}$ transfer functions. The discretization of these will be performed by one of the other available c2dm methods. One of these is the Tustin method and

another is the prewarp method. The Tustin method is also known as the bilinear method and the trapezoidal method. This method maps the left hand (stable) side of the continuous s-plane exactly into the unit circle (stable) region of the z-plane. The Tustin method substitutes (75) for **s** in the continuous transfer functions:

$$s = \frac{2}{T_s} \cdot \frac{z-1}{z+1} \qquad (2.4.4\text{-}1)$$

where:

z = unit delay, one sample time, T_s

T_s = sampling time (sec)

This substitution maps low frequencies into approximately the same digital frequencies, but produces a highly non-linear mapping of the high frequencies, (Dote, 1998, p. 127). However, since the entire $j\omega$ axis of the s-plane is mapped to the circumference of the unit circle in the z-plane, there is potential for distortion at high frequencies. To correct this distortion, a frequency prewarping scheme is used before the bilinear transformation. The frequency prewarping operation results in matching the signal critical frequency between the s-domain and the discrete z-domain. To achieve this result, the prewarping operation replaces **s** in the continuous transfer functions with $(\omega_o/\omega_p)s$, where ω_o is the frequency to be matched in the digital transfer function. Then ω_p is given by (2.4.4-2), (Dote, 1998, p. 127) and (Franklin, 1998, p. 194):

$$\omega_p = \frac{2}{T_s}\tan\left(\frac{\omega_o T_s}{2}\right) \qquad (2.4.4\text{-}2)$$

The selection of this critical frequency poses another tradeoff. The choices include the open loop gain crossover frequency, the disturbance frequency, the center frequency between a known range of frequencies, the filter cutoff frequency, to name a few. The objective is to derive discrete DOB filters that match the continuous system. By "match" it is implied that the continuous and discrete open loop frequency response predictions are in close agreement; i.e., the performance as predicted by open loop transfer function (2.3.4-1). To accomplish this, the continuous and discrete elements of the DOB controller will be matched and compared. Therefore, since discrete filters are notorious for adding phase lag to the system, the prewarp frequency selected is the gain crossover frequency, since it also determines the phase margin (PM), to insure that the discrete and continuous systems have approximately the same PM at the crossover frequency. The effect of prewarping can only be approximated using this pure linear analysis (with consideration of numerical saturation effect of the amplifier). The Bode plots for the third order $Q_{31}(s)$ DOB using prewarping at the gain crossover frequency (17.465 rad/s) are shown in Figure 2.17, for Q, 1 – Q, and Q/G_n transfer functions. Excellent matching is noted in the magnitude and phase plots of all of the plots in the set, especially at lower frequency (below 200 rad/s). This indicates that the

performance of the discrete DOB design is a close approximation to the continuous design. Similar results were obtained for all of the DOB algorithms studied. These plots can be reviewed in Appendix C.

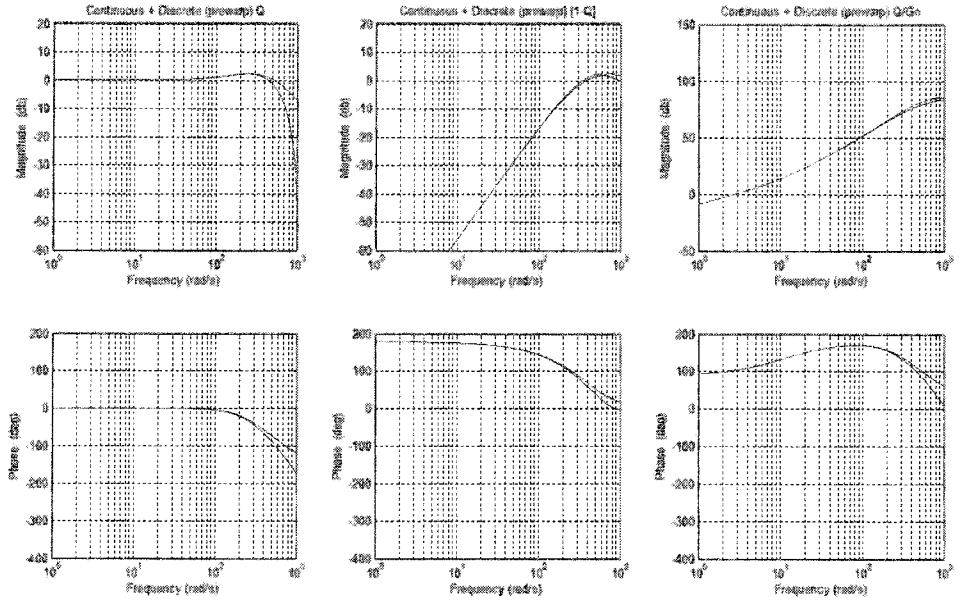

Figure 2.17: Bode Plot $Q_{31}(s)$ DOB
Continuous (blue) + Discrete (red) using prewarp at 17.465 rad/sec

2.4.4 Discrete Closed Loop Simulation Results

Figure 2.18a shows discrete simulation results for the closed loop SCARA system (Figure 2.16) to a 1.0 radian step input with both DOB and disturbance turned off. The results of the discrete simulation for the modified PD controller are very similar

to those of the continuous simulation. Settling time remains at ~1.8 seconds, and steady-state error is well below 0.005 rad.

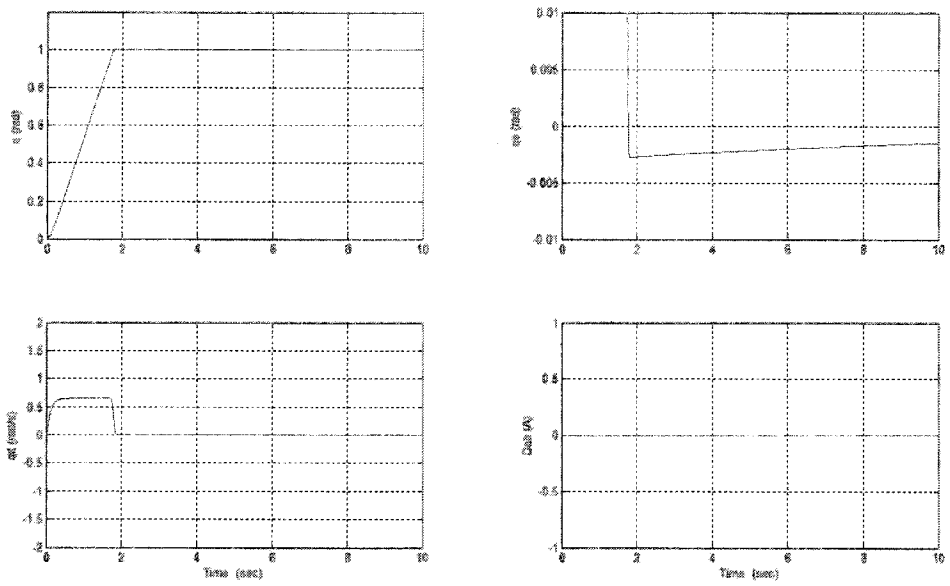

Figure 2.18a: Discrete Closed-Loop Response
DOB and Disturbance Off

Figure 2.18b shows simulation results for the same system as above to a 1.0 radian step input with a 0.5 amp disturbance. The DOB is again switched off in this simulation. As in the continuous simulation, the disturbance results in an angular position amplitude in excess of -0.035 rad, or 2.01°. The goal of the DOB algorithm designs is to improve on this result.

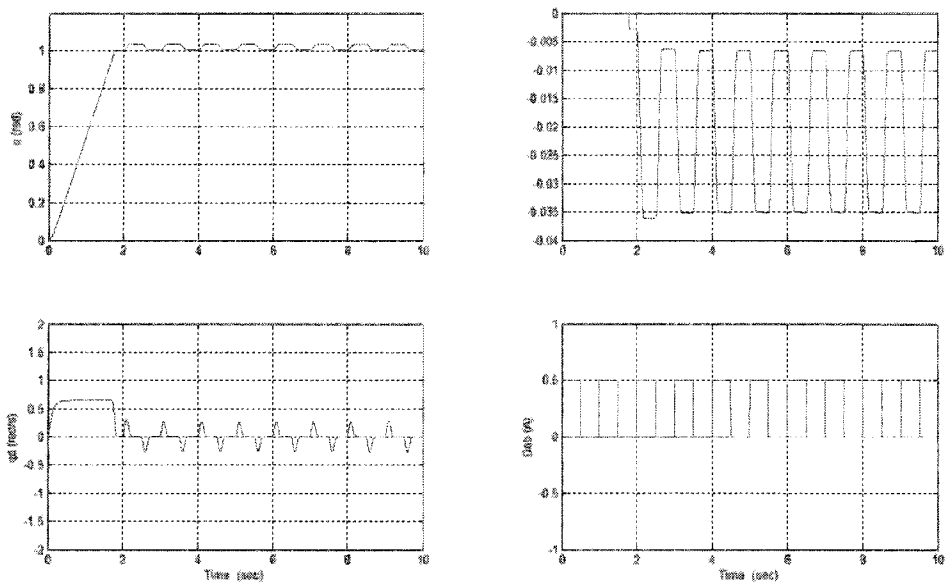

Figure 2.18b: Discrete Closed-Loop Response
DOB Off and Disturbance On

Figures 2.19 and 2.20 show the results of the Kaneko and the $Q_{20}(s)$ Filter DOB algorithms with the DOB implemented, respectively. As can be seen by comparing these two figures, the $Q_{20}(s)$ Filter DOB (Figure 2.20) shows a delay in the angular position response of the SCARA elbow link. The extra damping in this DOB algorithm probably accounts for most of the delay in the response.

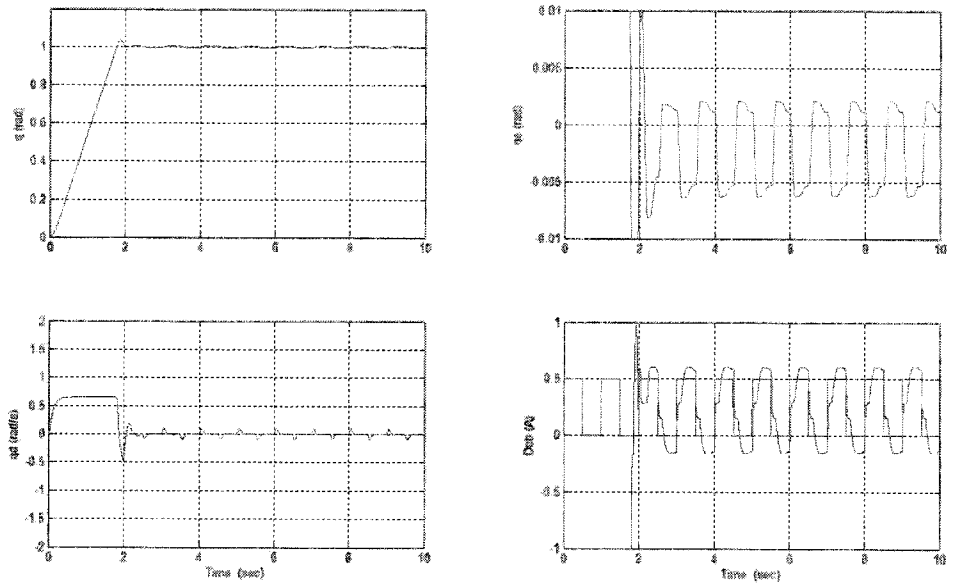

Figure 2.19: Discrete Time Simulation
$Q_{20}(s)$ Filter DOB (Kaneko)
f_c = 37Hz; α = 0.90; prewarp at 17.645 rad/sec

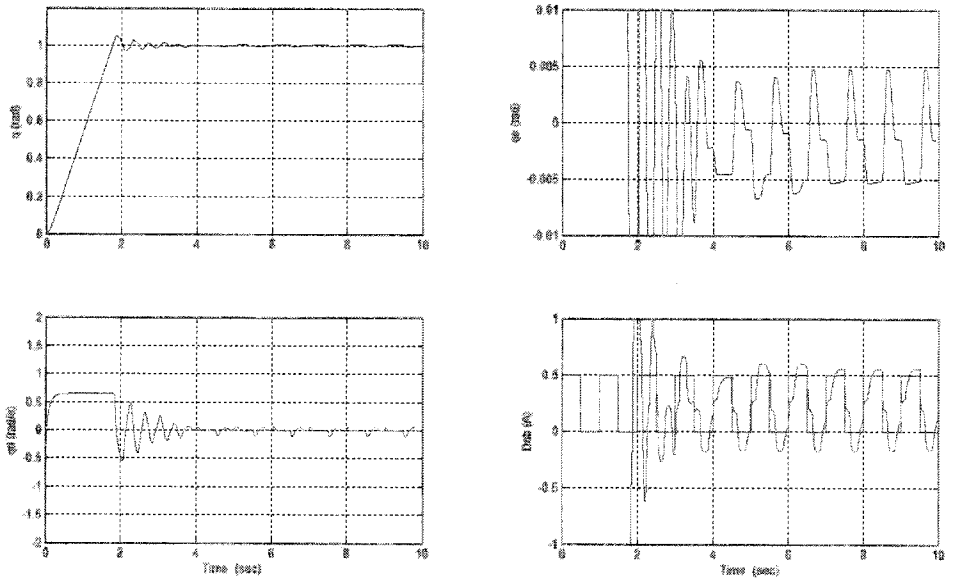

Figure 2.20: Discrete Time Simulation
$Q_{20}(s)$ Filter DOB with damping
f_c = 37Hz; α = 0.90; prewarp at 17.645 rad/sec

Since the performance of the above two DOB algorithms is at or above expectations, the third order $Q_{31D}(s)$ algorithm will be implemented. The discrete simulation using the $Q_{31D}(s)$ DOB algorithm is shown in Figure 2.21 for an alpha of 0.90.

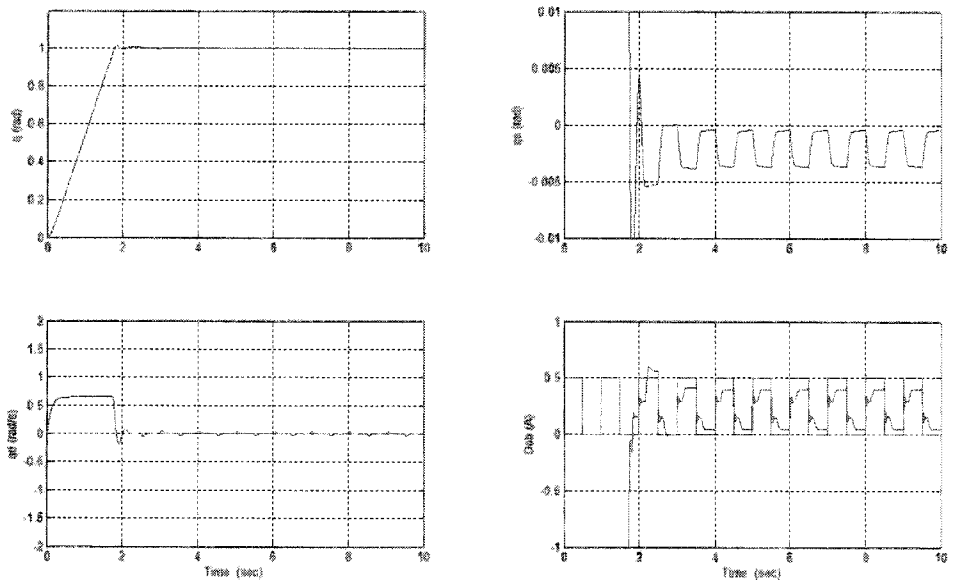

Figure 2.21: Discrete Time Simulation
$Q_{31D}(s)$ DOB
f_c = 44 Hz; α = 0.90; prewarp at 17.645 rad/sec

Now compare the open loop response of the prewarped digital system (Figure 2.22) to that of the continuous system (Figure 2.8). It is necessary to verify that the discrete model performs similarly in open loop when compared to the continuous time analysis performed in Section 2.3.4. This study determined the gain and phase margins for the open loop system, including the effect of the DOB

and α. The gain and phase margins are GM = 86.2 db at 2717 rad/sec and PM = 34.1° at 17.465 rad/sec, respectively, and are independent of α.

A second Matlab script (dob_design4z.m, Appendix B.2) similar to the previous, determined the gain and phase margins for the discrete case. The gain and phase margins for the discrete case also show very little dependence on α (as shown in Figure 2.22 and tabulated in Table 2.2).

Table 2.2: Relationship between Alpha vs. GM and PM for Discrete Time

α	**Gain Margin**	**Phase Margin**
1.0	42.9 db at 524 rad/s	34.3° at 17.5 rad/s
0.95	45.4 db at 507 rad/s	34.2° at 17.5 rad/s
0.90	48.2 db at 466 rad/s	34.0° at 17.5 rad/s
0.85	47.1 db at 488 rad/s	34.1° at 17.5 rad/s

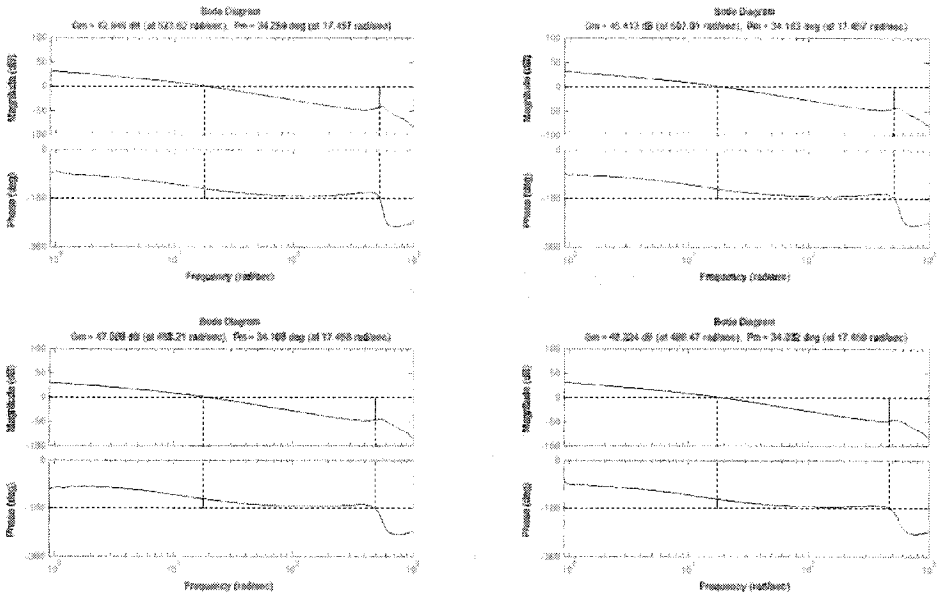

Figure 2.22: Open Loop Gain and Phase Margin – Discrete Time
(Clockwise from upper left, α = 1.0, 0.95, 0.90, 0.85)

A final analytical study is performed to demonstrate the robustness of the DOB algorithm. In this study, the Simulink model shown in Figure 2.16 is used. The DOB algorithm $Q_{31D}(s)$ is used with an alpha value of 0.90 at a cutoff frequency of 44 Hz. Combinations of the effective inertia, $n_I \cdot J_{EFF}$, and the effective Coulomb friction, $n_{CF} \cdot F_{EFF}$, are varied as shown in Figure 2.23, until as wide a box as could be realized is obtained while maintaining acceptable performance. The viscous friction, B_{EFF}, was held at its nominal value for this study. However some liberty was taken with regard to the percent overshoot. This parameter was allowed to attain a level of 30 – 40% since it can be easily reduced by softening the step input, or by delaying the onset of the DOB algorithm. With this assumption in

place, the DOB algorithm can deliver good error results for large variations in these SCARA mechanical properties.

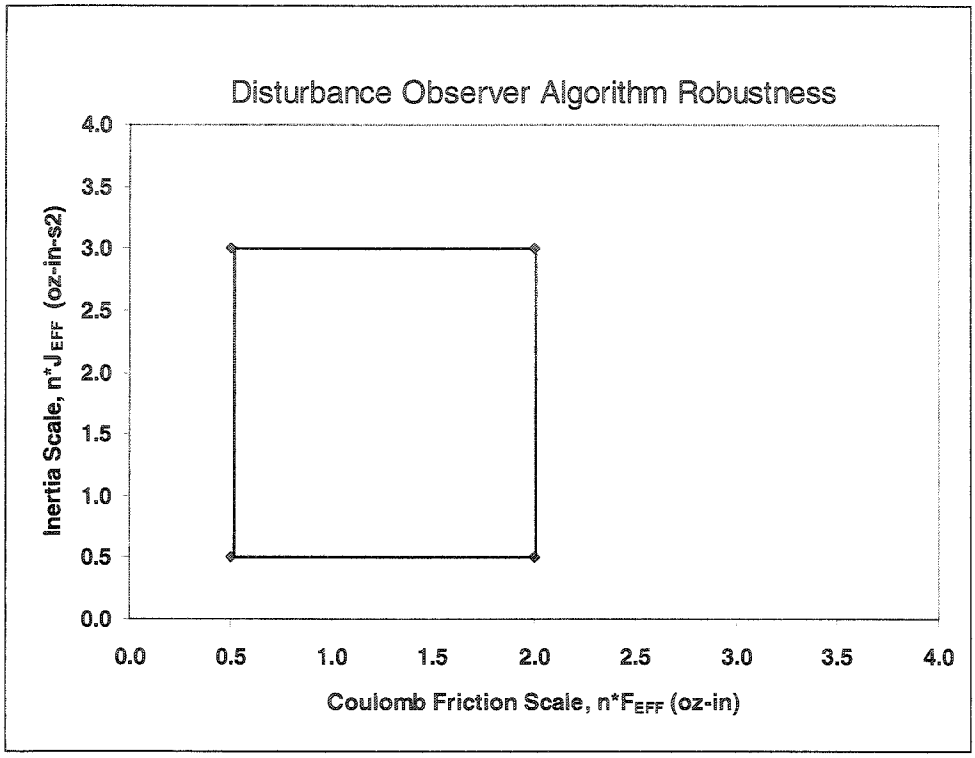

Figure 2.23: Disturbance Observer Algorithm Robustness
The $Q_{31D}(s)$ DOB algorithm is used to determine the performance of the SCARA for a range of inertias and Coulomb friction. The DOB controller can be effective over a fairly wide range in parameter variability.

Chapter 3

Implementation

The implementation of the DOB is performed on the SJSU SCARA (as shown in Figure 3.1). Only the elbow link will be used in the experiments. The shoulder link was held rigid during testing.

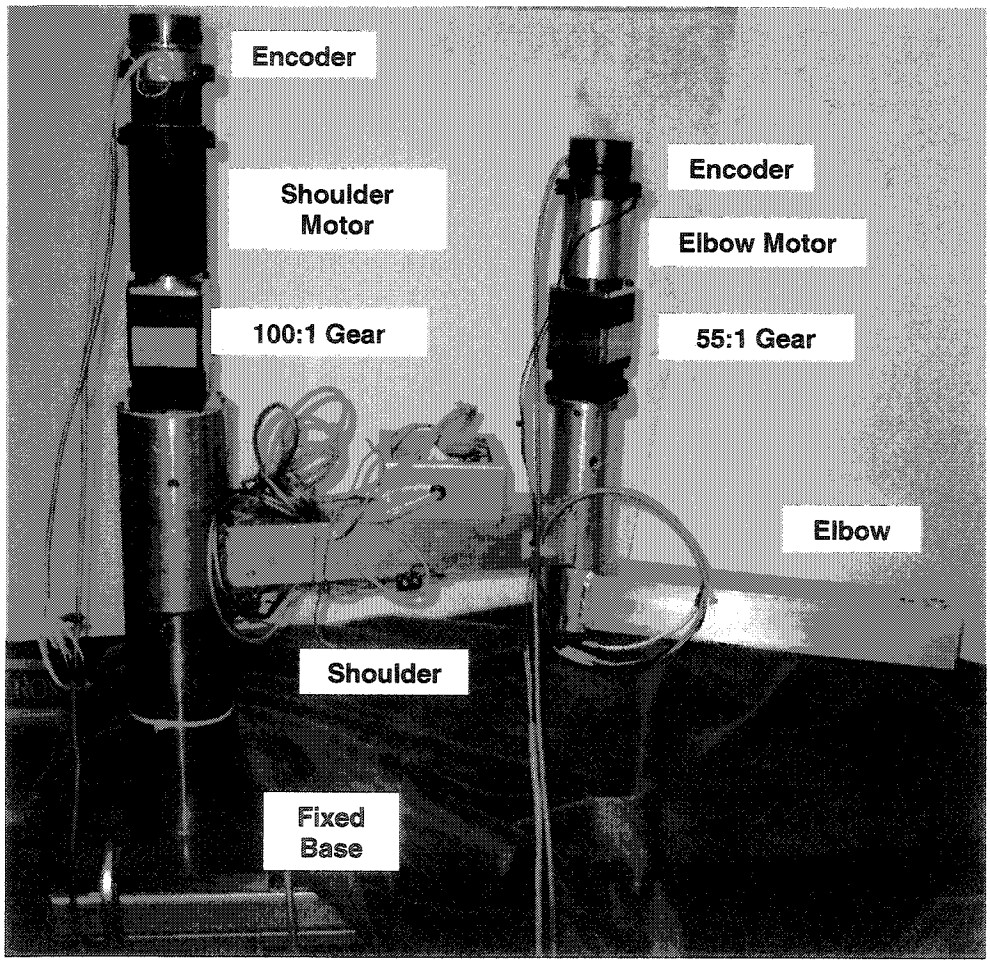

Figure 3.1: SJSU SCARA

3.1 SCARA Test Setup

The SJSU SCARA test setup is shown in Figure 3.2. The mechanical structure of the robot consists of a thick-wall cylindrical base, and two linkages (the shoulder link, and the elbow link). Each motor assembly includes a planetary gearhead attached to the rotor, and an encoder at its base to measure the angular position of the link. The shoulder link is constructed of thick-wall aluminum bar welded between two cylinders, one large at the shoulder end, and to another (smaller diameter) thick-wall cylinder at the elbow end. This second cylinder is the mount for the elbow motor, planetary gearhead, and encoder. So to clarify, both motors are rigidly attached to the upper part shoulder link (on top of the cylinders. The shoulder motor and gear assembly output shaft extends through its cylinder to drive against the fixed base. The elbow motor and gear assembly output extends through its own cylinder and is attached to the elbow link. The output gear of the shoulder gearhead is fixed to the base, so that motor torque causes the motor and gearhead housing to rotate. Conversely, the output gear of the elbow motor and gearhead rotates the elbow link against a motor and gearhead housing fixed to the shoulder link.

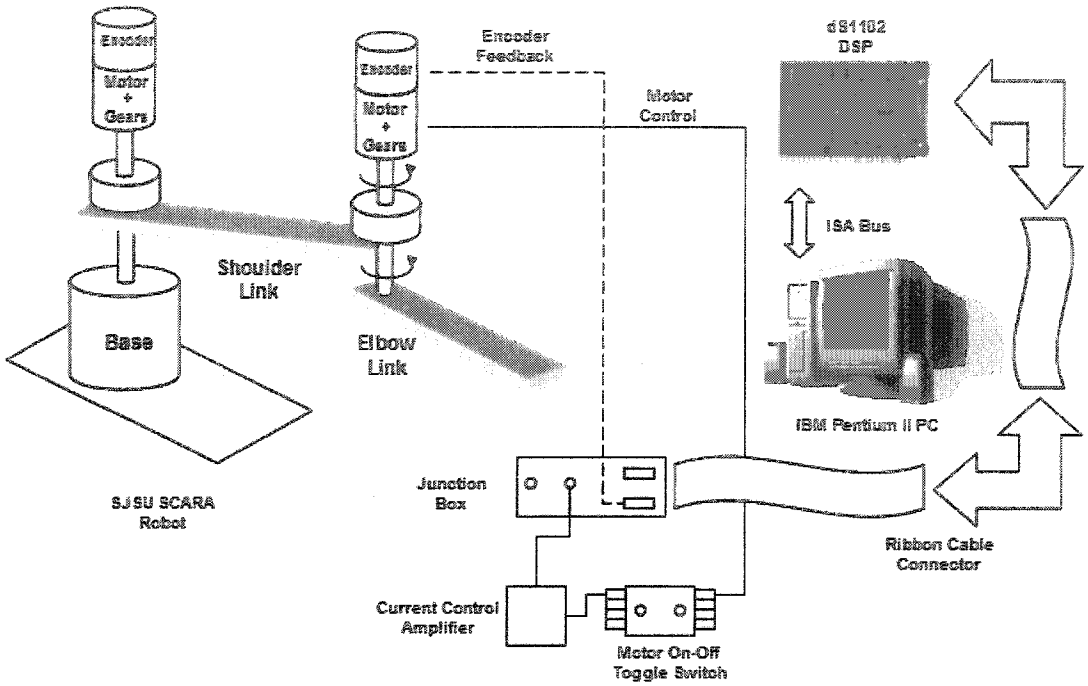

Figure 3.2: SJSU SCARA Test Setup

The SCARA electronics consists of two electronic amplifiers (one high gain amplifier for the shoulder link, and one low gain amplifier for the elbow link), and a digital signal processor (DSP) board, Figure 3.2. Lastly, a junction box supplied by dSpace© takes motor control data from a ribbon cable connected to the DSP board, and also puts the encoder feedback onto this cable to go back to the DSP. The input voltage to the amplifier must be limited to protect the motors (and to some extent personnel…at higher voltage the arms make a full rotation in less than one second). The DSP board has a gain called the amplifier or D/A gain

(0.077 A/V for the elbow). The output of the D/A is an analog current which is sent to the motor coils to develop torque. The reference or "digital" currents (I* and I_{d_hat} in Figure 3.3) are converted to a "digital" voltage via the amplifier gain. This voltage is then bounded in software to protect the motor. So a repeat of the gains accounted for in software must now be accounted for in the model of the hardware and firmware.

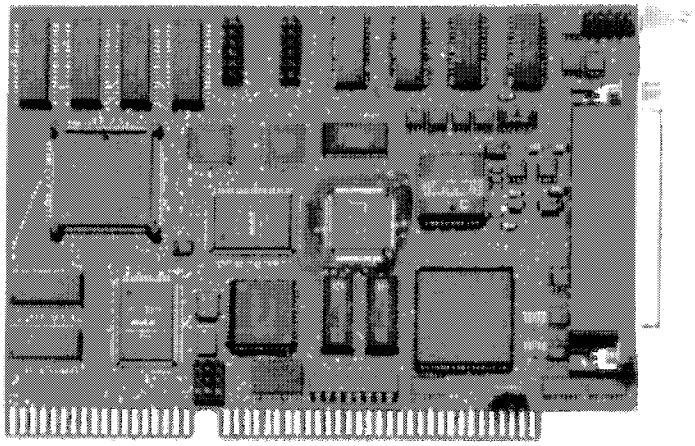

Figure 3.3: dSpace© ds1102 DSP Controller Board

An IBM PC with a Pentium 2 (200 MHz) processor is used as the interface to the SCARA. The dSpace© ds1102 DSP board is located in one of the PC expansion slots.

3.2 Discrete Closed Loop Implementation Model

The Simulink model shown in Figure 3.4 is the actual code used to interface between the designer and the DSP board. This code is converted into the C language and then compiled into object code, by executing a "Build" command in Matlab/Simulink RealTime Workshop (which takes a little less than one minute for this model). A typical run procedure is documented in Appendix D.

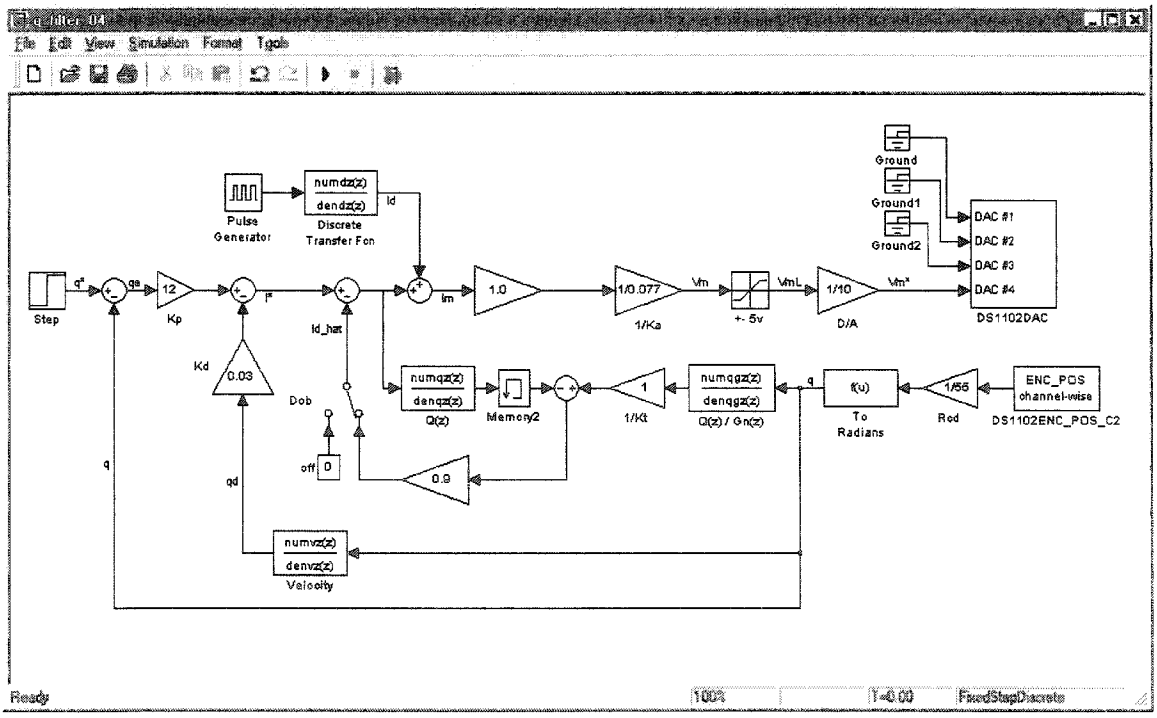

Figure 3.4: Matlab / Simulink model for Elbow Robotic Arm

Most of the block diagram used for continuous and discrete analysis has been utilized for the implementation of the DOB on the SCARA. A few blocks have

been added to interface between the SCARA and software in the forward loop as well as the feedback loop. Of course the model of the SCARA dynamics is replaced by the actual SCARA hardware.

An incremental optical encoder measures angular position feedback. This type of encoder takes whatever position at the start of a run as its zero position. dSpace© provides the encoder blocks (and others) to be used in the Simulink model. There are actually two different encoder blocks supplied. One block reads and outputs encoder position as a scaled output (±1 in Simulink corresponding to ±1 revolution), which can be converted to radians or degrees. The other block outputs a measured delta angular position between successive encoder lines. In this case since the encoder is attached to the DC motor at the base, the measured angular delta position is of the motor rotor. This measurement can be used for angular velocity feedback by dividing the converted and scaled reading by the sampling time used by the digital controller. This method of determining the velocity was traded with an alternative method using the angular position feedback and the transfer function of (2.1.3-2) used in all of the analysis. The trade consisted of an open loop Simulink model with both shoulder and elbow links (as shown in Figure 3.6). The test shows the comparison. Overall comparison showed very little difference between the measurements. Close inspection showed a slightly reduced noise content in the filter version (q2dz). Filters however generally contribute to the overall phase lag. It was determined that choosing the signal which produced the lower noise content would better serve the

implementation of the DOB. In retrospect, the direct angular velocity measurement may be the better choice. The velocity measurement is not smooth even at steady-state operation, a sign that some wear may be occurring in the joint bearings or other location.

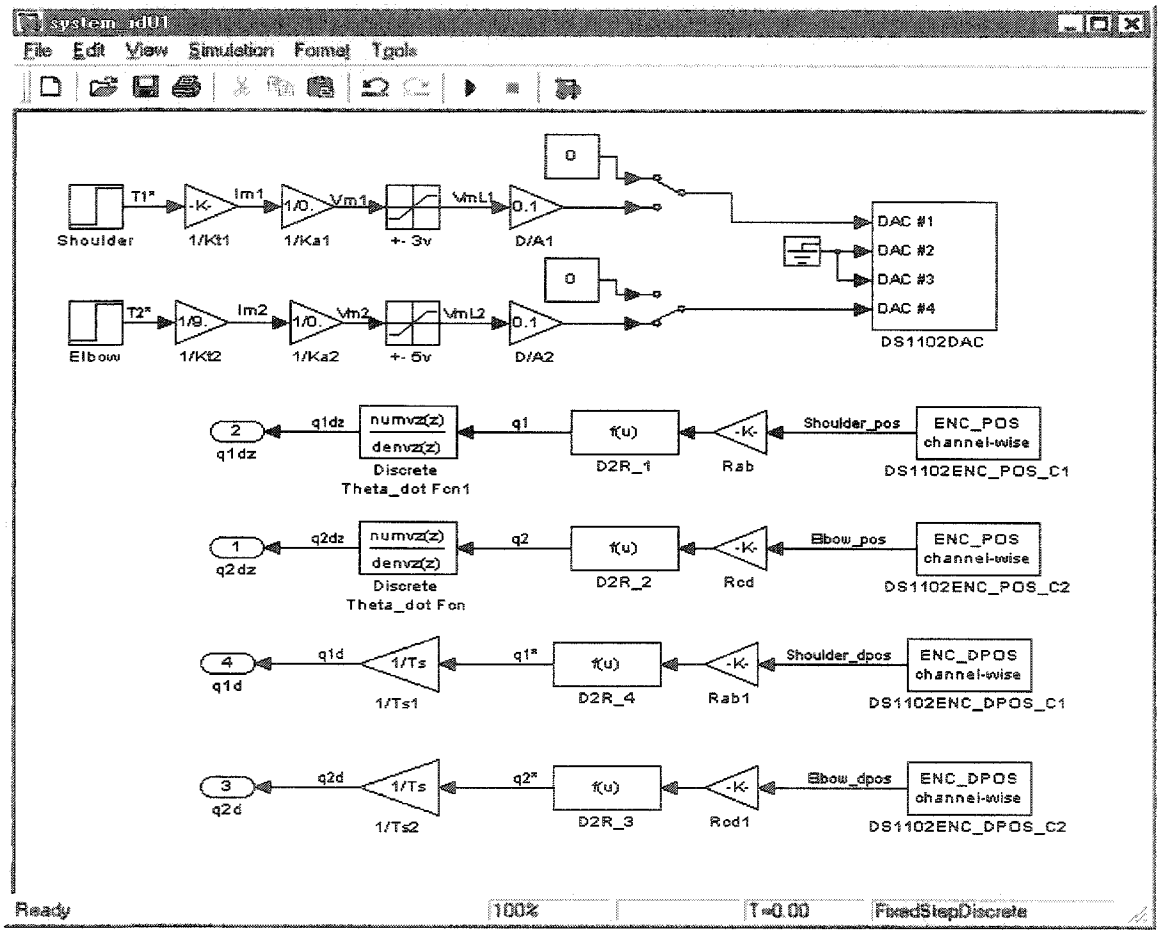

Figure 3.5: Simulink Model to Compare Encoder Velocity vs. Transfer Function Velocity

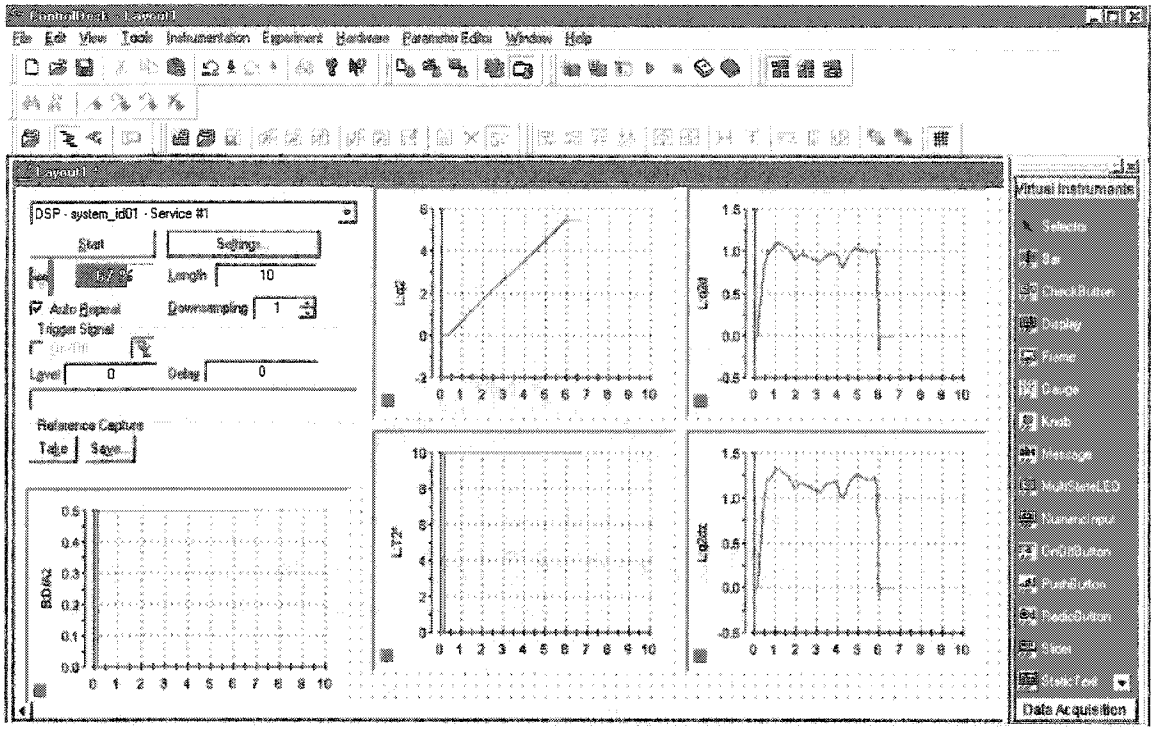

Figure 3.6: Raw test data showing results of the Encoder Test
Clockwise from upper left: Angular position feedback (q2, rad), velocity via delta position encoder measurement (q2d, rad/s), velocity via delta position encoder measurement (q2d, rad/s), Torque input (T_2, oz_f-in).

The encoder measurement is made using a 24 bit word size and a resolution of 1024 encoder lines per revolution. Each encoder channel uses fourfold pulse multiplication for enhanced resolution (4096 equivalent encoder lines). One bit of the 24 bit signal is used as a sign bit leaving 23 bits for the signal. The encoder angular resolution can be determined from this data as slightly less than 0.1°:

$$\text{resolution} = \frac{2\pi \dfrac{\text{rad}}{\text{rev}}}{4 \cdot \left(1024 \dfrac{\text{encoder lines}}{\text{rev}} \right)} = 0.00153 \text{ rad } (0.0879°) \qquad (3.2\text{-}1)$$

The encoder sends a scaled value between -1 and 1 back to Simulink. This value must be converted to radians prior to use in the feedback control loop. The resulting conversion from the raw encoder output to radians is given by (3.2-1). Note that the raw encoder output is scaled to the motor rotor position. It must first be divided by the gear ratio since the link angular position is the desired feedback.

The digital scaling of the encoder measurement must also be accounted for. As an example, reduce the number of bits in the digital word from 23 to 3 plus the sign bit. Then the range of the measured parameter would be divided over $2^3 = 8$ values. This is shown graphically in Figure 3.7. Since the angular range in this application is -2π to $+2\pi$ radians, each vertical step ($\Delta\theta = \pm 2\pi/2^3$ rad = ±0.7854 rad) corresponds to a digital code from 000 to 111 as shown in the figure. Therefore a digital reading of 101 would correspond to 3.927 rad. Strictly speaking, since the $+2\pi$ and -2π positions correspond to the same encoder reading, the resolution should be $\Delta\theta = \pm \dfrac{2\pi}{2^3 - 1}$ rad = 0.8976 rad, which is a significant difference from the above calculation (de Silva, 1989, p. 228). However, when the measurement is scaled using 23 bits the difference will be negligible.

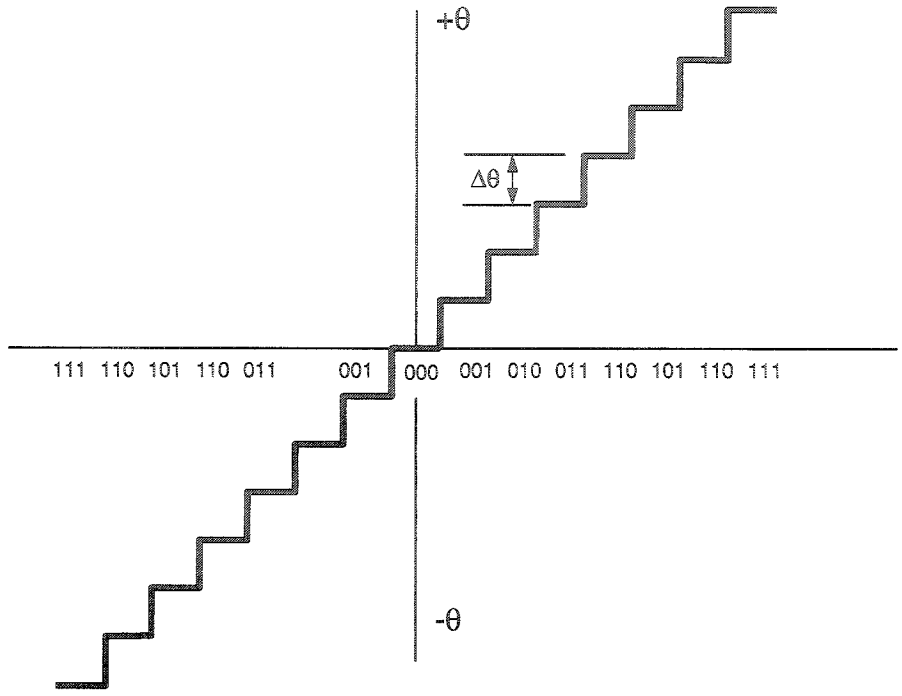

Figure 3.7: Encoder Digital Resolution
(An additional sign bit is used to distinguish positive and negative output codes)

The two scaling factors are combined into a single scale factor in the implementation model given by (3.2-2) which returns angular position in radians.

$$\theta(\text{rad}) = 2^{23} \cdot \frac{2\pi \frac{\text{rad}}{\text{rev}} \cdot \left(\begin{matrix} \text{encoder} \\ \text{scaled output} \end{matrix} \right)}{4\left(1024 \frac{\text{encoder lines}}{\text{rev}} \right)} = 2^{21} \cdot \frac{2\pi \frac{\text{rad}}{\text{rev}} \cdot \left(\begin{matrix} \text{encoder} \\ \text{scaled output} \end{matrix} \right)}{1024 \frac{\text{encoder lines}}{\text{rev}}} \quad (3.2\text{-}2)$$

One other new block has been added. In the middle of the DOB between the output of the Q(s) filter and the DOB summing junction, a memory block has been

inserted. An algebraic loop is created by the prewarp algorithm. This block provided the best performance by breaking the algebraic while adding a time delay of only one third of a sample period or less. Time delay is to be avoided at all costs due to its exponential contribution to phase lag. There may be a better way to implement an algebraic loop breaker which introduces zero phase lag, but this will have to be a planned upgrade in a subsequent thesis.

Lastly, the input control signal from Simulink to the dSpace© ds1102 board must be scaled by a factor of 10. This is to counteract the same scale factor (in reverse) imposed by the dSpace© board, hence, the D/A gain in Figure 3.7.

3.3 ControlDesk© Experiment Design

ControlDesk© is the software supplied by dSpace© used to interface between the operator, Simulink, and the DSP ds1102 board. A detailed set of steps necessary to run an experiment is given in Appendix D. The design of the layout in this case is merely for functionality; i.e., to record enough quick look data to assess DOB performance. In addition the layout allows the option of storing data from the various designated channels to a Matlab *.mat file for analysis. Potentially, the software will allow certain parameters to be tuned within an experiment run without recompiling. Unfortunately, this required more RAM than was available. Since only about 30 to 40 seconds were needed to recompile, this was not a major obstacle.

The plots are laid out for each experiment as shown in Figure 3.8, with reference also to the Simulink block diagram shown in Figure 3.4.

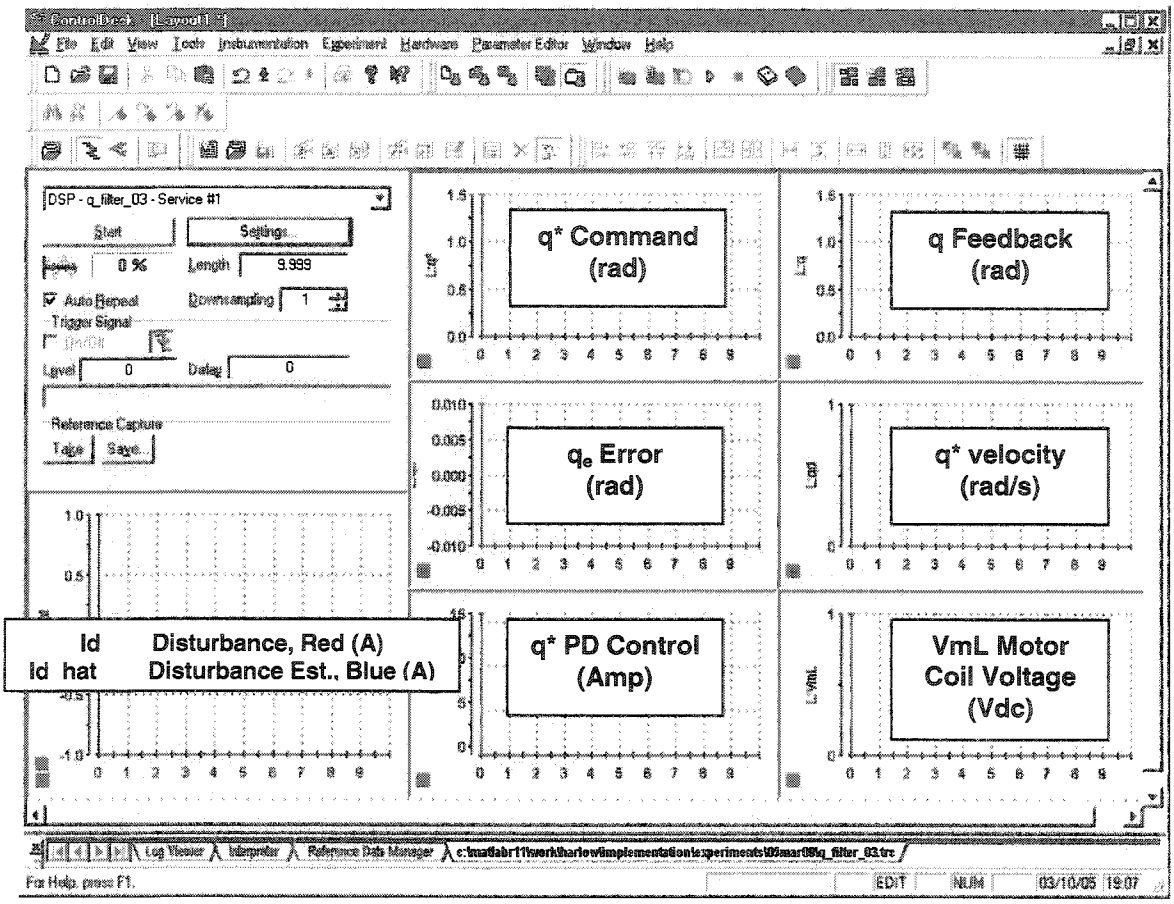

Figure 3.8: dSpace© ControlDesk© Experimentation Layout

One major obstacle was corrected during the course of the experimentation. dSpace© plots were being clipped (the first 2 to 4 sec of data was lost in each run). One workaround was found through trial and error: the step input in Simulink was set to start at 3.5 sec. This way the time delay (which may well be a RAM issue) only lost trivial data.

3.4 Test Results

The results of a test matrix consisting of one alpha value ($\alpha = 0.9$) and three cutoff frequencies (40 Hz, 60 Hz, and 80 Hz) for each of the four DOB algorithms is presented. The four algorithms are: $Q_{20}(s)$ without damping, $Q_{20}(s)$ with damping, $Q_{31D}(s)$ with damping, and $Q_{42D}(s)$ with damping. The test results are compared in Table 3.1 for peak-to-peak steady-state error. A qualitative assessment of the motor control voltage is also noted in the body of the text. The test data plots for the 60 Hz cutoff frequency case are presented in this section.

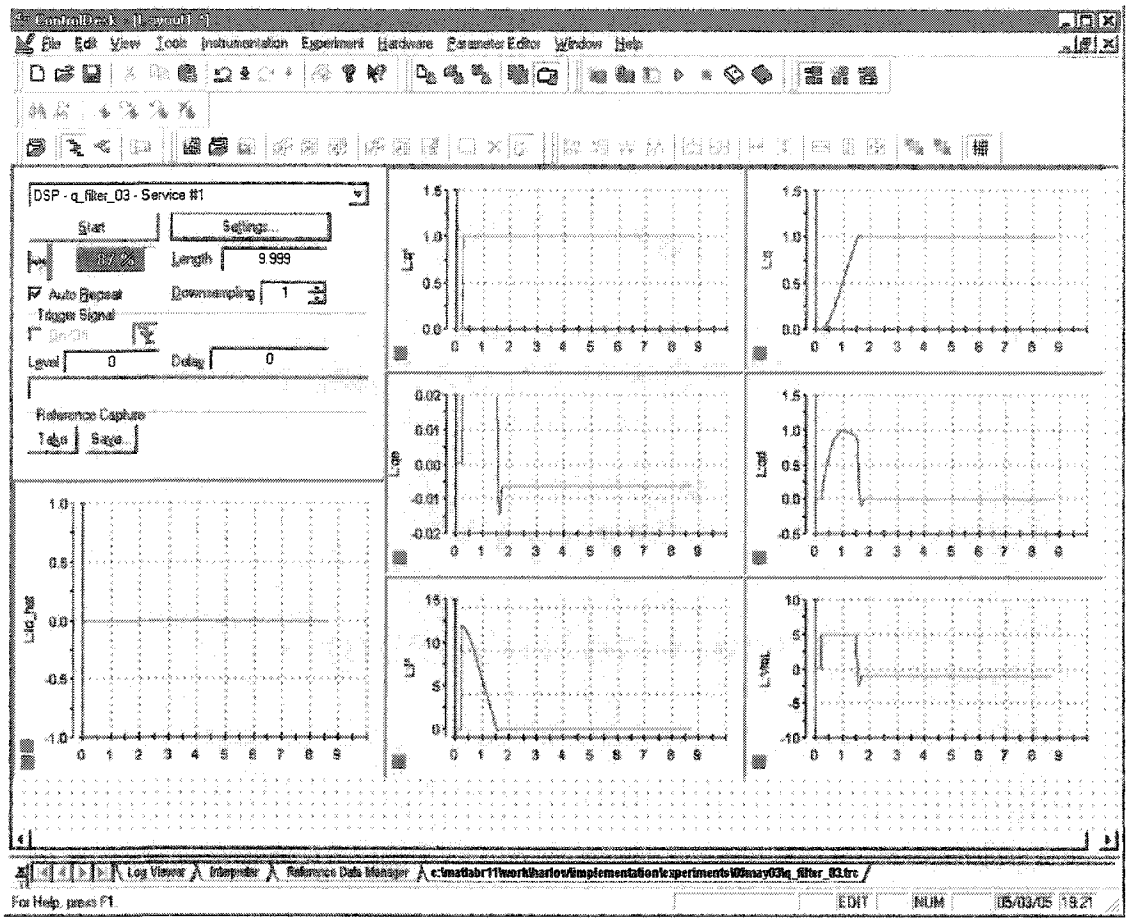

Figure 3.9: dSpace© Run with Disturbance Off and DOB Off

The first experiment to be analyzed is one with both disturbance and disturbance observer turned off. From this data it can be seen that the outer PD control loop is well tuned (Figure 3.9). As shown by the angular error signal, q_e, the steady-state error is approximately 0.006 rad (0.3°). It is also apparent that the motor coil voltage is saturated at 5vdc for most of the time that the elbow is in slewing to the command set point (1 rad).

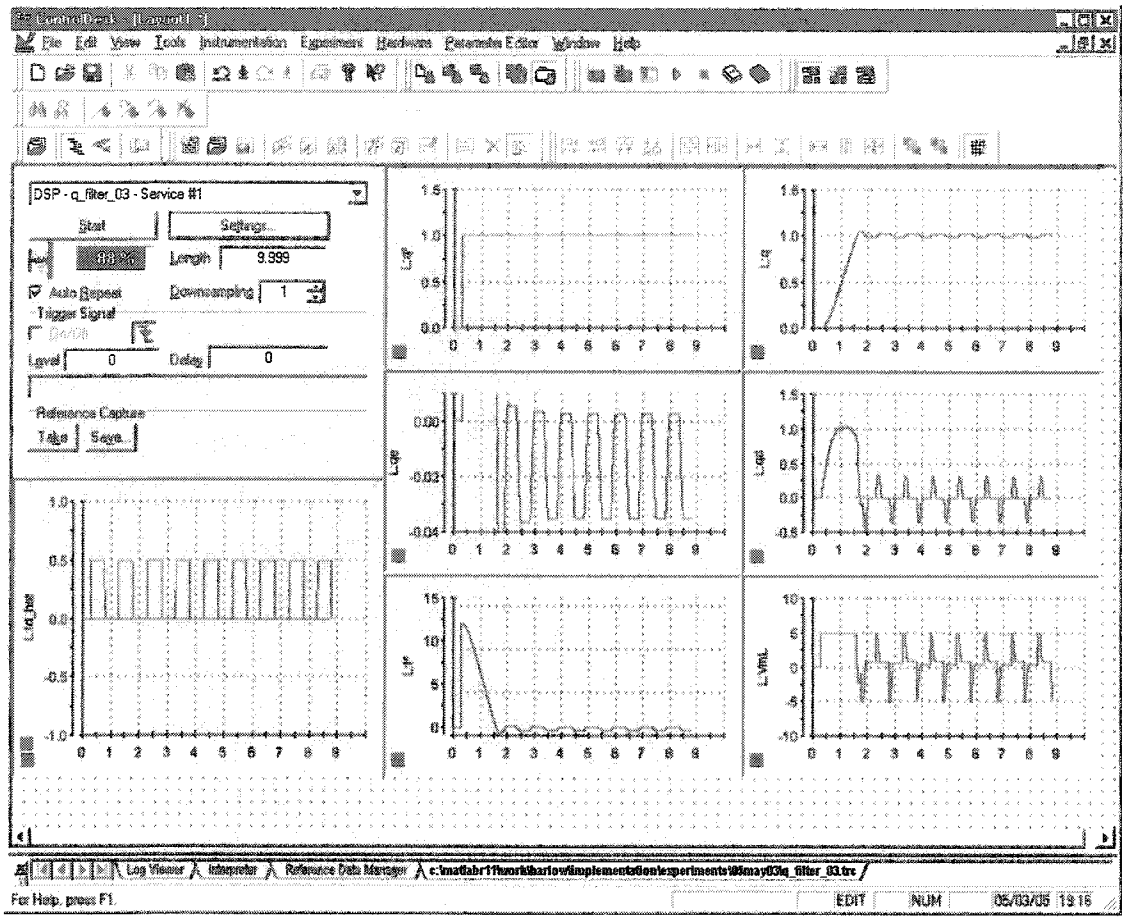

Figure 3.10: dSpace© Run with Disturbance On and DOB Off

Next the effect of adding the pulse train (0 to 0.5 A) disturbance is analyzed. Figure 3.10 clearly shows the ripple pulses in the disturbance (I_d), shown in Red in the lower left plot, feedback (q) and the steady-state error (q_e). The peak-to-peak magnitude of the steady-state error ranges is 0.040 rad (2°).

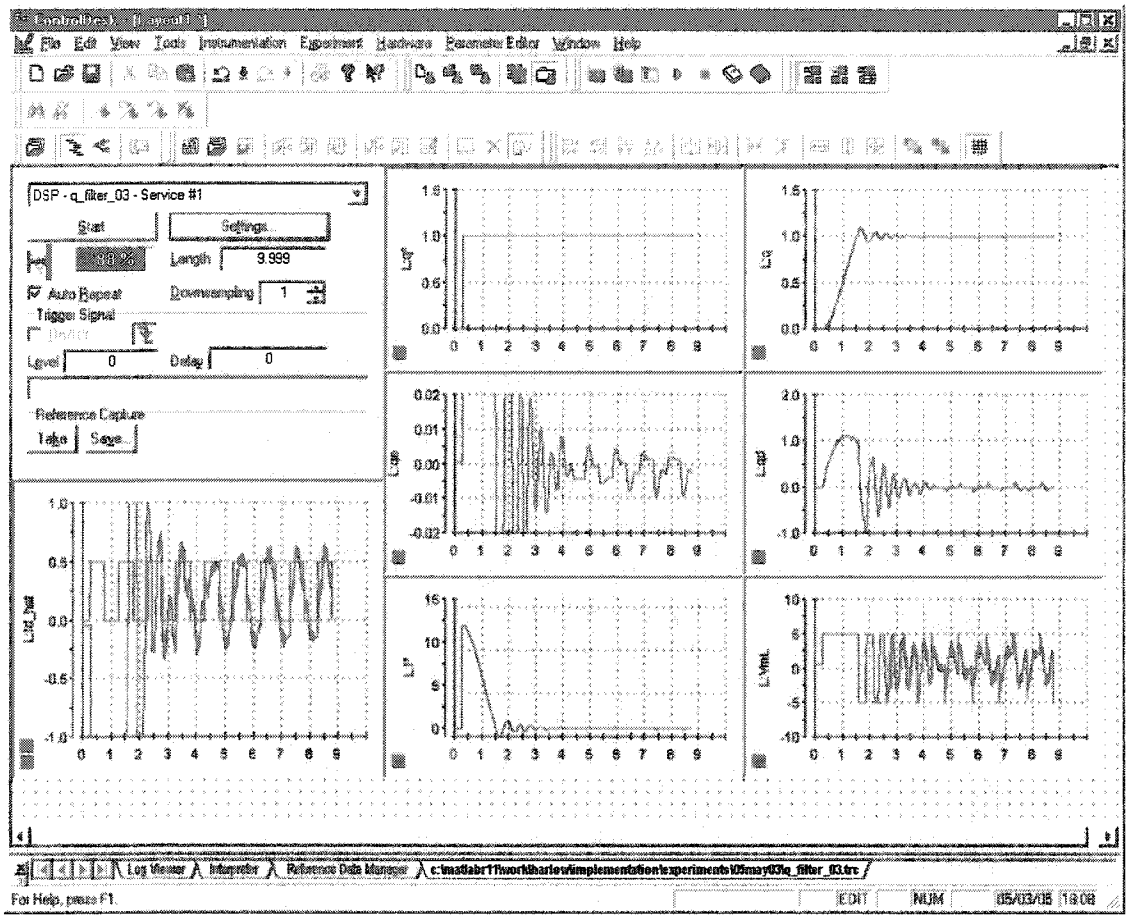

Figure 3.11: dSpace© Run with Q_{20}, no Damping (Kaneko Algorithm)
f_c = 60 Hz; α = 0.90

Finally, the effect of the disturbance observer can be shown. Figure 3.11 shows the $Q_{20}(s)$ DOB algorithm without damping (Kaneko Algorithm). This run used a cutoff frequency of 60 Hz and an alpha value of 0.90. The overshoot is about 10%, the peak-to-peak steady-state error is now reduced to 0.011 rad (0.63°). The disturbance estimate (blue) is showing a fair reproduction of the actual disturbance (red) in the lower left plot.

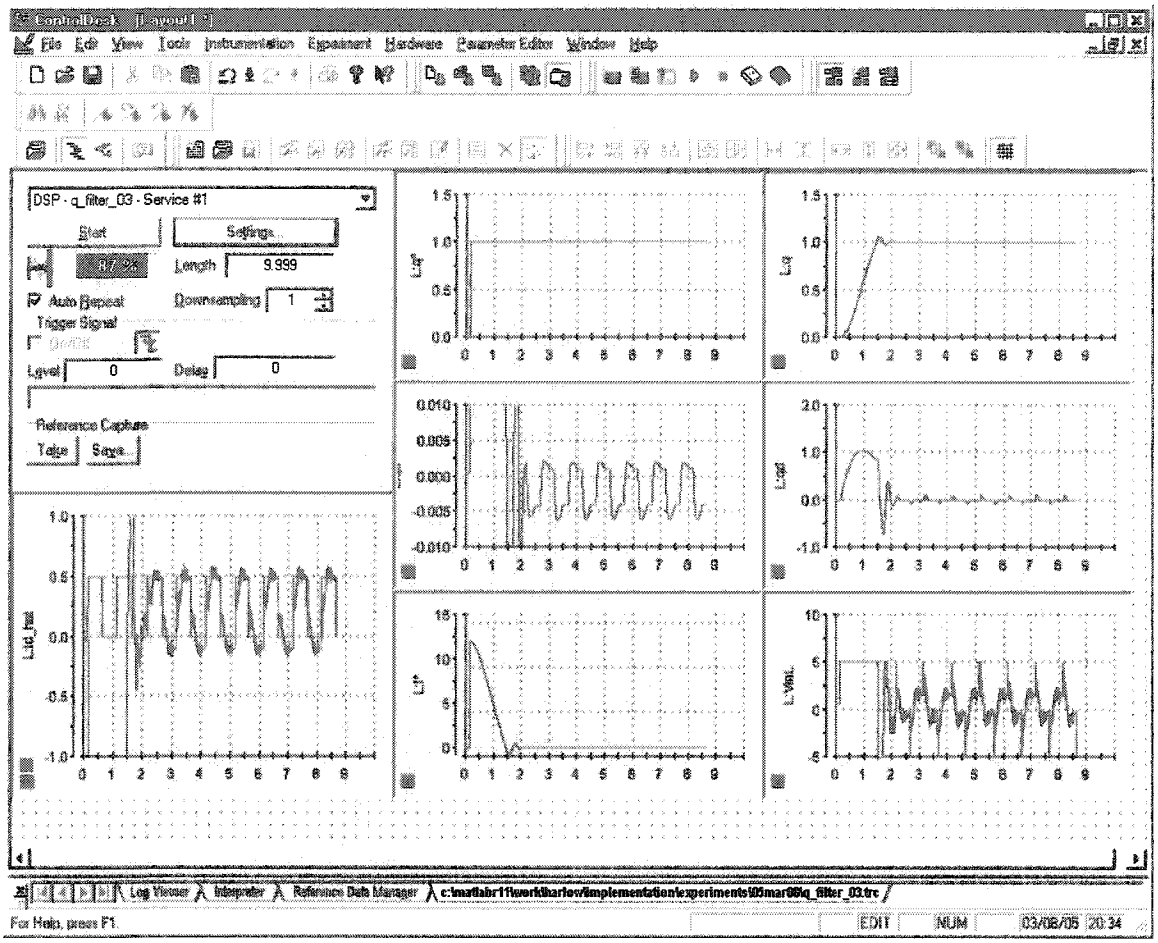

Figure 3.12: dSpace© Run with Q_{20} with Damping
f_c = 60 Hz; α = 0.90

Figure 3.12 shows the $Q_{20}(s)$ DOB algorithm with damping for a cutoff frequency of 60 Hz and an alpha value of 0.90. The overshoot is now <10%, the peak-to-peak steady-state error is now 0.009 rad (0.5°). The disturbance estimate (blue) is showing an improved reproduction of the actual disturbance (red). This is likely due to the experimental determination of the viscous friction coefficient used in the DOB nominal model.

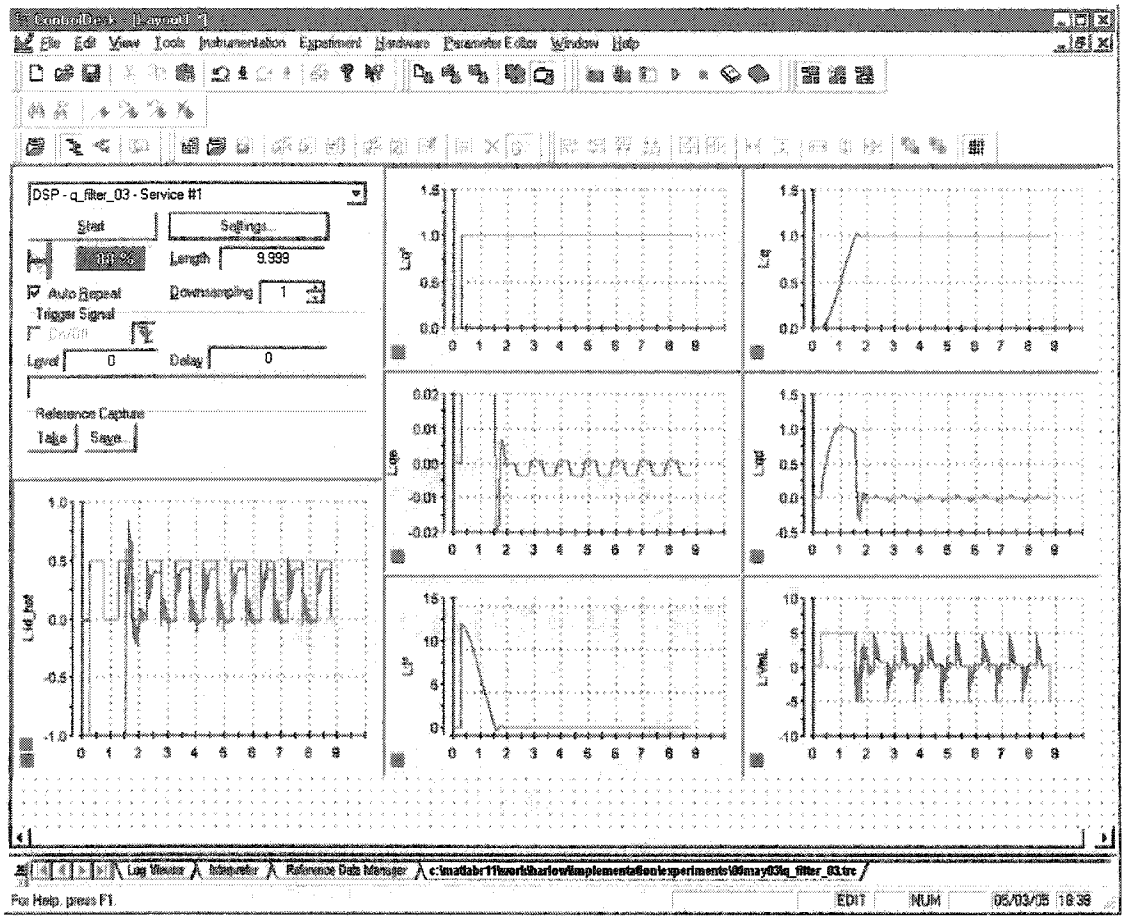

Figure 3.13: dSpace© Run with Q_{31D} with Damping
f_c = 60 Hz; α = 0.90

Figure 3.13 shows the $Q_{31D}(s)$ DOB algorithm with damping. This run used a cutoff frequency of 60 Hz and an alpha value of 0.90. The overshoot is <5%, and dampens rapidly. The peak-to-peak steady-state error is now 0.004 rad (0.2°). This observation is consistent with the predicted disturbance reduction potential for the third order Q(s) compared to the second order filter as shown in the 1-Q(s) magnitude plot of Figure 2.7.

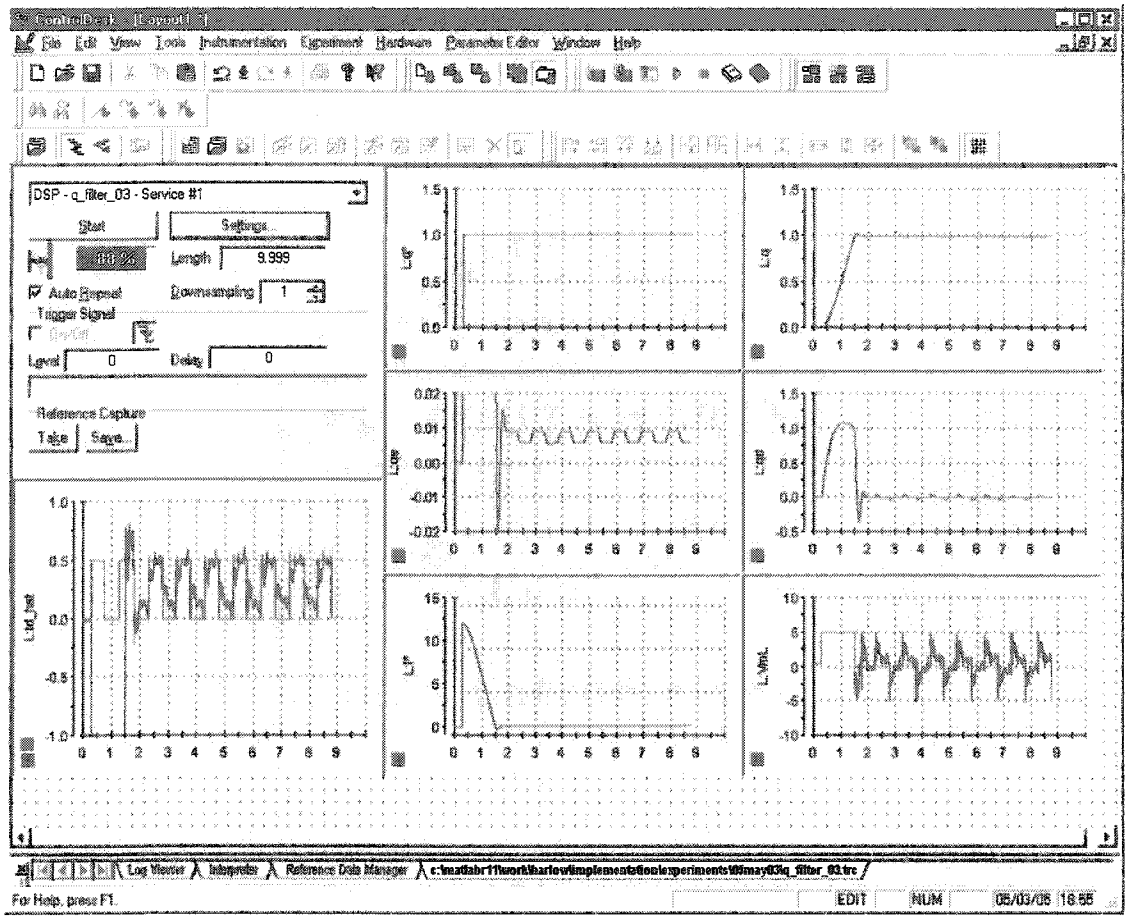

Figure 3.14: dSpace© Run with Q_{42D} with Damping
f_c = 60 Hz; α = 0.90

Figure 3.14 shows the $Q_{42D}(s)$ DOB algorithm with damping. This run used a cutoff frequency of 60 Hz and an alpha value of 0.90. The overshoot is <5%, and dampens rapidly. The peak-to-peak steady-state error is now reduced to 0.003 rad (<0.2°). Again an improvement is observed in disturbance rejection with this fourth order DOB algorithm as predicted by Figure 2.7.

The results of the above test runs and additionally two other filter cutoff frequencies are shown in Table 3.1. Although theory predicts improvement in disturbance rejection for the higher order filters presented, it was believed that certain negating effects would have had more influence. For example, the effect of the DOB performance on phase margin for the overall system and the additional computational burden on the DSP as the filter order increases.

Table 3.1: Test Results

DOB Algorithm	Peak-to-Peak Steady-State Error		
	f_c = 40 Hz	f_c = 60 Hz	f_c = 80 Hz
$Q_{20}(s)$ No Damping	0.012 rad	0.011 rad	0.009 rad
$Q_{20}(s)$ with Damping	0.010	0.009	0.008
$Q_{31}(s)$ with Damping	0.003	0.004	0.003
$Q_{42}(s)$ with Damping	0.002	0.003	0.003

Different trends in steady-state error are noted in the above test results. The second order filter performance is improved when damping is included in the DOB algorithm. There is a marked improvement in steady-state error for the third and fourth order filters. These DOB algorithms reduced the 0.040 rad disturbance down to 0.002 to 0.003 rad. However, there is a slight increase in voltage activity as the cutoff frequency is creased from 40 Hz to 80 Hz. Surprisingly little variation is noted for each of the filters when the cutoff frequency is decreased from 60 Hz to 40 Hz or increased to 80 Hz.

Chapter 4

Conclusions

The objective of this thesis is to develop the disturbance observer algorithm and implement it using the SJSU SCARA robot. Four DOB algorithms using three different Q(s) filters with orders from two to four are compared and contrasted.

The effect of using a nominal model consisting only of inertia in the estimate of the disturbance is also demonstrated in conjunction with a second order Q(s) filter. The inertia only model greatly reduced the steady-state error effect of the disturbance pulse train from 0.040 rad down to 0.009 rad for a cutoff frequency of 80 Hz. Some means of reducing the effect of the overshoot and subsequent ringing may be necessary. It is concluded based on the experimental data for the second order filter that acceptable results may be realized for applications when viscous friction estimates are not possible.

It is demonstrated that better steady-state error performance is obtained for the higher order DOB algorithms presented in this thesis. These DOB algorithms reduced the 0.040 rad disturbance down to 0.002 to 0.003 rad. However for all of the DOB algorithms, there is a small increase in motor coil voltage activity as the cutoff frequency increases. It is still believed that it is good design practice to select the lowest order filter which delivers acceptable results.

4.1 Recommendations

Several studies are recommended for follow-on effort.

1. Implement a PID outer loop controller in place of the modified PD controller to eliminate steady-state offset due to asymmetrical disturbances.
2. Perform an in-depth study of measurement noise effect on DOB performance, and contrast with higher frequency disturbances.
3. Perform a more rigorous stability analysis; in particular the effect of the saturation power amplifier.
4. Perform a more rigorous robustness (sensitivity to varying effective inertia, non-linear friction, etc.) analysis or experimentation.
5. Perform a friction experiment to quantify the effects of stiction, viscous and Coulomb friction.
6. Perform an experiment to determine the gain crossover frequency.
7. Extend the implementation of the DOB to the two arm SCARA configuration.

References

Craig, J. J. (1989) . *Introduction to robotics mechanics and control.* (2nd ed.), Reading, Massachusetts: Addison-Wesley-Longman Publishing Company.

Dang, K. T. (2003) . *Application of disturbance observer for force estimation and motion control.* Unpublished master's thesis, San Jose State University, San Jose, CA.

de Silva, C. W. (1989) . *Control sensors and actuators.* Englewood Cliffs, NJ: Prentice-Hall.

Dote, Y., & Hoft, R. (1998) . *Intelligent control power electronic systems.* New York: Oxford University Press.

Franklin, G., & Powell, J. & Workman, M. (1998) . *Digital control of dynamic systems.* (3rd ed.), Menlo Park, California: Addison-Wesley-Longman Publishing Company.

Franklin, G., & Powell, J., & Emami-Naeini, A. (2002) . *Feedback control of dynamic systems.* (4th ed.), Upper Saddle River, NJ: Prentice-Hall.

Kaneko, K., et. al., (1994) . *Manipulator control based on a disturbance observer in the operational space.* Proc. IEEE Int. Conference, Robotics and Automation, Vol. 2, (pp 902-909).

Kempf, C. & Kobayashi, S., (1999) . *Disturbance observer and feed-forward design for a high speed direct drive position table.* IEEE Transactions on Control Systems Technology, Vol. 7, No. 5, (pp. 513-526).

Raven, F. H. (1995) . *Automatic control engineering (Fifth ed.)* . New York, NY: McGraw-Hill, Inc.

Shahian, B. & Hassul, M., (1993) . Control *system design using Matlab.* Englewood Cliffs, NJ: Prentice-Hall.

Tai, Ta-Jen, (2001). dSpace© based motion control of robotic manipulators. Unpublished master's thesis, San Jose State University, San Jose, CA.

Umeno, T., & Hori, Y., (1991) . *Robust speed control of dc servomotors using modern two degrees-of-freedom controller design.* IEEE Trans. Ind. Electron., Vol. 38, No. 5, (pp. 363–368).

Webopedia computer dictionary. (n.d.). Retrieved April 23, 2005, from http://www.pcwebopedia.com/TERM/D/DVD.htm

White, M., et. al., (1999). *Rejection of disk drive vibration and shock disturbances with a disturbance observer.* Proceedings of the American Control Conference, San Diego, CA., (pp. 4127 – 4131).

Appendices

Appendix A

SJSU SCARA Parameters

Parameter	Description	Value	Units
I	Inertia, elbow	0.2063	lb_f-in-s^2
J_m	Inertia, Motor Rotor, elbow	0.004	oz_f-in-s^2
J_G	Inertia, Gearhead, elbow	0.233	oz_f-in-s^2
M_m	Mass, Motor + Gearhead, elbow	5.0	lb_m
m	Mass, elbow	2.086	lb_m
L	Length, elbow	11.75	In
L_c	Length, elbow – Pivot to cg	4.64	in
N_g	Gear Ratio, elbow	55	-
F_{EFF}	Coulomb Friction Coefficient, elbow	1.632	oz_f-in
B_{EFF}	Viscous Friction Coefficient, elbow	3.438	$\frac{oz_f - in - s}{rad}$
K_t	Motor Torque Constant, elbow	9.6	$\frac{oz_f - in}{A}$
K_e	Motor Back EMF Constant, elbow	0.0678	$\frac{V - s}{rad}$
L_m	Motor Coil Inductance, elbow	2.0	mH
R_m	Motor Coil Resistance, elbow	1.38 - 1.87	Ω
τ_{Tme}	Motor Electrical Time Constant, elbow (L_m/R_m)	1.25	msec
τ_m	Motor Mechanical Time Constant, elbow (J_m/B_m)	9.53	msec
G_a	Amplifier Gain, elbow	0.077	$\frac{A}{V}$

Appendix B

Matlab m-Files

B.1 dob_design4c.m

```
% Open Loop Design - Continuous
%
% D. Harlow & Dr. J.C. Wang
% SJSU, 1-24-05

Jn    = 0.318;
Bn    = 3.438;
kp    = 12;
kd    = 0.03;
alf   = 0.98;

% PD Controller
numv = [1 0];
denv = [0.020 1];

GC1    = kp;
GC2    = kd*tf(numv,denv);

% 3rd order Q31 DOB
% Q-Filter
fc    = 80;
wc    = 2*pi*fc;        % cutoff frequency, rad/s
t     = 1/wc;
numq = [3*t 1];
denq = [t^3 3*t^2 3*t 1];
GQ    = tf(numq,denq);

% Q/Gn
numqg = [3*Jn*t (3*Bn*t+Jn) Bn 0];
denqg = 9.6*[t^3 3*t^2 3*t 1];
GQGN  = tf(numqg,denqg);

% Plant - SCARA
numgp = 9.6;
dengp = [0.318 3.438 0];
GP    = tf(numgp,dengp);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Subplot(221)
alf = 1.0;

GLSN  = GC1*GP;
GLSD  = 1 - alf*GQ + alf*GP*GQGN + GC2*GP;
```

```
GLOS   = GLSN/GLSD;
[ncl,dcl] = tfdata(GLOS,'v');

% Plant Stability
%w      = [0.01:0.05:1000];
w      = [0.9:0.1:1000];                                %
[mag,ph] = bode(GLOS,w);
margin(mag,ph,w)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Subplot(222)
alf = 0.95;

GLSN   = GC1*GP;
GLSD   = 1 - alf*GQ + alf*GP*GQGN + GC2*GP;
GLOS   = GLSN/GLSD;
[ncl,dcl] = tfdata(GLOS,'v');

% Plant Stability
w      = [0.9:0.1:1000];                                %
[mag,ph] = bode(GLOS,w);
margin(mag,ph,w)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Subplot(223)
alf = 0.90;

GLSN   = GC1*GP;
GLSD   = 1 - alf*GQ + alf*GP*GQGN + GC2*GP;
GLOS   = GLSN/GLSD;
[ncl,dcl] = tfdata(GLOS,'v');

% Plant Stability
w      = [0.9:0.1:1000];                                %
[mag,ph] = bode(GLOS,w);
margin(mag,ph,w)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Subplot(224)
alf = 0.85;

GLSN   = GC1*GP;
GLSD   = 1 - alf*GQ + alf*GP*GQGN + GC2*GP;
GLOS   = GLSN/GLSD;
[ncl,dcl] = tfdata(GLOS,'v');

% Plant Stability
w      = [0.9:0.1:1000];                                %
[mag,ph] = bode(GLOS,w);
margin(mag,ph,w)
```

B.2 dob_design4z.m

```
% Open Loop Design - Discrete
%
% D. Harlow & Dr. J.C. Wang
% SJSU, 1-24-05

Jn    = 0.318;                              % Elbow inertia,
Bn    = 3.438;                              %
kp    = 12;                                 %
kd    = 0.03;                               %
Ts    = 0.003;                              %

% PD Controller
numv = [1 0];                               %
denv = [0.020 1];                           %
[numvz,denvz]=c2dm(numv,denv,Ts);           %

GC1    = kp;                                %
GC2    = kd*tf(numvz,denvz,Ts);             %

% 3rd order Q31 DOB
% Q-Filter
fc    = 80;                                 %
wc    = 2*pi*fc;                            % cutoff frequency, rad/s
t     = 1/wc;                               %
numq = [3*t 1];                             %
denq = [t^3 3*t^2 3*t 1];                   %
[numqz,denqz]=c2dm(numq,denq,Ts,'prewarp',17.465);       %
GQ    = tf(numqz,denqz,Ts);                 %

% Q/Gn
numqg = [3*Jn*t (3*Bn*t+Jn) Bn 0];          %
denqg = 9.6*[t^3 3*t^2 3*t 1];              %
[numqgz,denqgz]=c2dm(numqg,denqg,Ts,'prewarp',17.465);   %
GQGN   = tf(numqgz,denqgz,Ts);              %

% Plant - SCARA
numgp = 9.6;                                %
dengp = [0.318 3.438 0];                    %
[numgpz,dengpz] = c2dm(numgp,dengp,Ts);     %
GP     = tf(numgpz,dengpz,Ts);              %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Subplot(221)
alf = 1.0;

% Gey
GLSN   = GC1*GP;                                        %
GLSD   = 1 - alf*GQ + alf*GP*GQGN + GC2*GP;             %
GLOS   = GLSN/GLSD;                                     %

% Plant Stability
```

```
w     = [0.9:0.1:1000];                                    %
[mag,ph] = bode(GLOS,w);                                   %
margin(mag,ph,w)                                           %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Subplot(222)
alf = 0.95;

% Gey
GLSN  = GC1*GP;                                            %
GLSD  = 1 - alf*GQ + alf*GP*GQGN + GC2*GP;                 %
GLOS  = GLSN/GLSD;                                         %

% Plant Stability
w     = [0.9:0.1:1000];                                    %
[mag,ph] = bode(GLOS,w);                                   %
margin(mag,ph,w)                                           %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Subplot(223)
alf = 0.90;

% Gey
GLSN  = GC1*GP;                                            %
GLSD  = 1 - alf*GQ + alf*GP*GQGN + GC2*GP;                 %
GLOS  = GLSN/GLSD;                                         %

% Plant Stability
w     = [0.9:0.1:1000];                                    %
[mag,ph] = bode(GLOS,w);                                   %
margin(mag,ph,w)                                           %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Subplot(224)
alf = 0.85;

% Gey
GLSN  = GC1*GP;                                            %
GLSD  = 1 - alf*GQ + alf*GP*GQGN + GC2*GP;                 %
GLOS  = GLSN/GLSD;                                         %

% Plant Stability
w     = [0.9:0.1:1000];                                    %
[mag,ph] = bode(GLOS,w);                                   %
margin(mag,ph,w)                                           %
```

B.3 q_filter_setup.m

```
% qfilter_setup
%
% D. Harlow
% SJSU


Ts    = 0.003;
numv = [1 0];
denv = [0.020 1];
[numvz,denvz]=c2dm(numv,denv,Ts);

Jn    = 0.318;
Bn    = 3.438;

%%%
```

```
%%%
```

Appendix C
Simulation Plots

Continuous Time Simulation Plots

Figure C.1: Continuous Simulation
3rd Order $Q_{31D}(s)$ DOB; f_c = 44 Hz; α = 0.90

Figure C.2: Continuous Simulation
4th Order $Q_{42D}(s)$ DOB; f_c = 68 Hz; α = 0.90

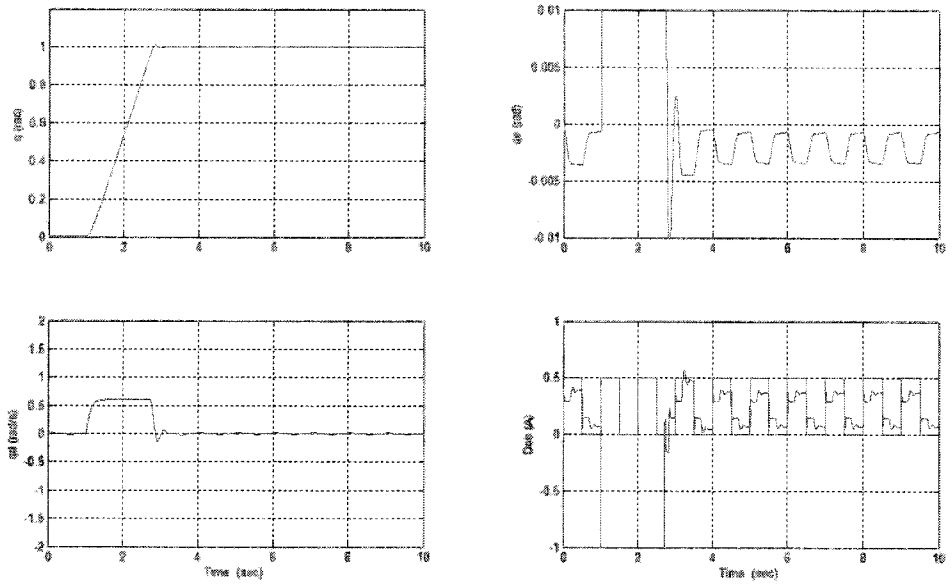

Figure C.1: Continuous Simulation
3rd Order $Q_{31D}(s)$ DOB f_c = 44 Hz; α = 0.90

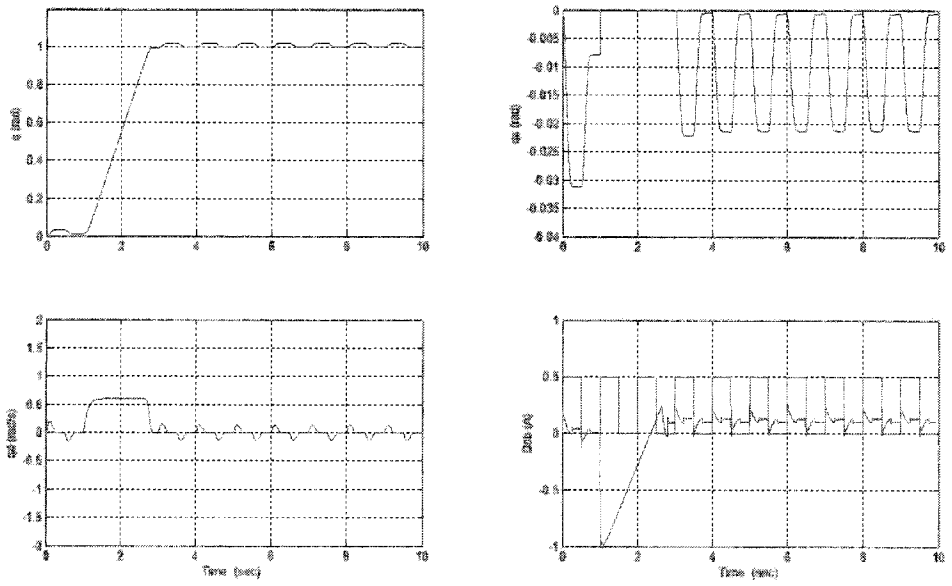

Figure C.2: Continuous Simulation
4th Order $Q_{42D}(s)$ DOB f_c = 68 Hz; α = 0.90

Discrete Time Simulation Plots

Figure C.3: Discrete Time Simulation
4^{th} Order $Q_{42D}(s)$ DOB
f_c = 68 Hz; α = 0.90; prewarp at 17.645 rad/sec

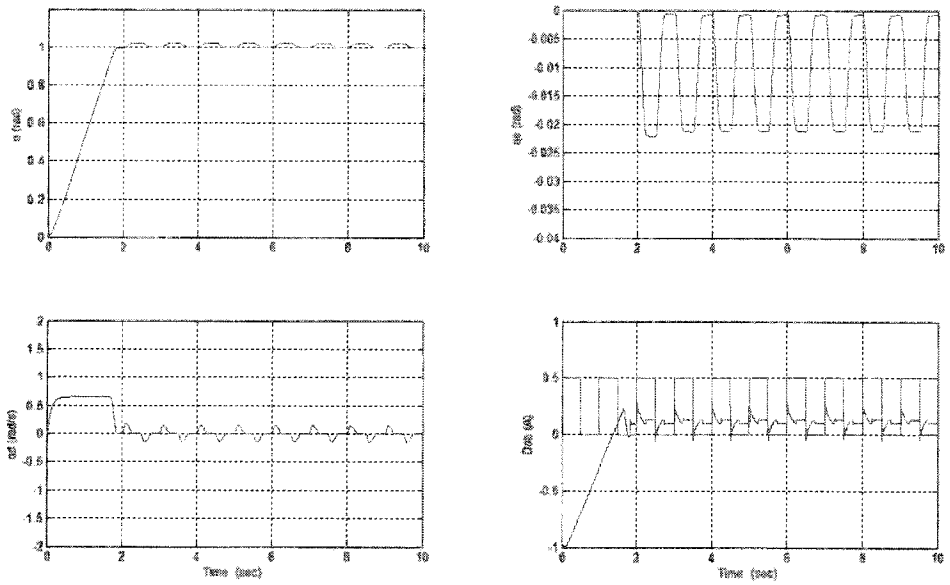

Figure C.3: Discrete Time Simulation
4^{th} Order $Q_{42D}(s)$ DOB
f_c = 68 Hz; α = 0.90; prewarp at 17.645 rad/sec

Bode Plots: Continuous vs. Discrete for Zero Order Hold and Prewarp

Figure C.4: **Bode Plots - Continuous vs. Discrete using Prewarp**
2^{nd} order $Q_{20}(s)$ Filter DOB (Kaneko)
f_c = 57Hz; prewarp at 17.645 rad/sec

Figure C.5: **Bode Plots - Continuous vs. Discrete using Prewarp**
2^{nd} order $Q_{20}(s)$ Filter DOB with damping
f_c = 57 Hz; prewarp at 17.645 rad/sec

Figure C.6: **Bode Plots - Continuous vs. Discrete using Prewarp**
3rd Order $Q_{31D}(s)$ DOB
f_c = 44 Hz; prewarp at 17.645 rad/sec

Figure C.7 **Bode Plots - Continuous vs. Discrete using Prewarp**
4^{th} Order $Q_{42D}(s)$ DOB
f_c = 68 Hz; prewarp at 17.645 rad/sec

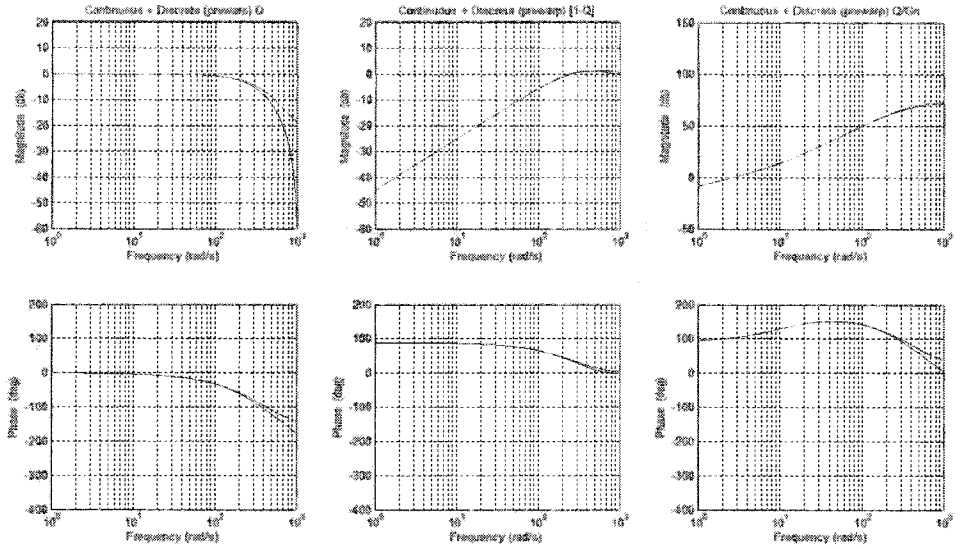

Figure C.4: **Bode Plots - Continuous vs. Discrete using Prewarp**
2^{nd} order $Q_{20}(s)$ Filter DOB (Kaneko)
f_c = 57Hz; prewarp at 17.645 rad/sec

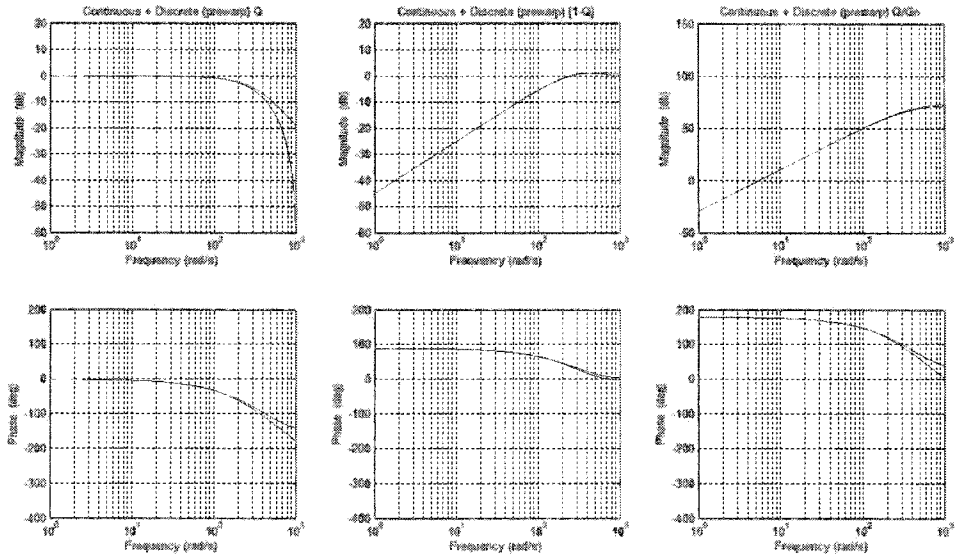

Figure C.5: **Bode Plots - Continuous vs. Discrete using Prewarp**
2^{nd} order $Q_{20}(s)$ Filter DOB with damping
f_c = 57 Hz; prewarp at 17.645 rad/sec

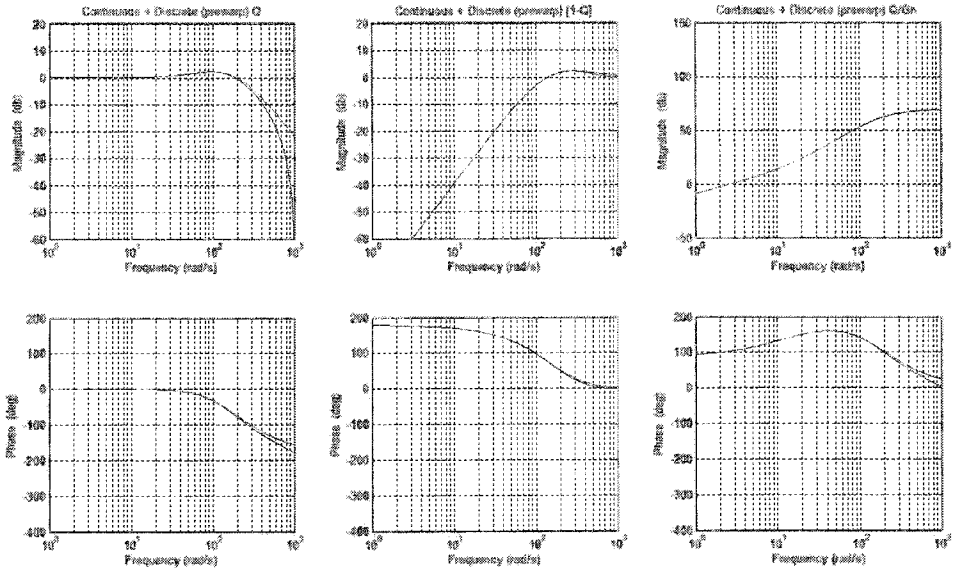

Figure C.6: **Bode Plots - Continuous vs. Discrete using Prewarp**
3rd Order $Q_{31D}(s)$ DOB
f_c = 44 Hz; prewarp at 17.645 rad/sec

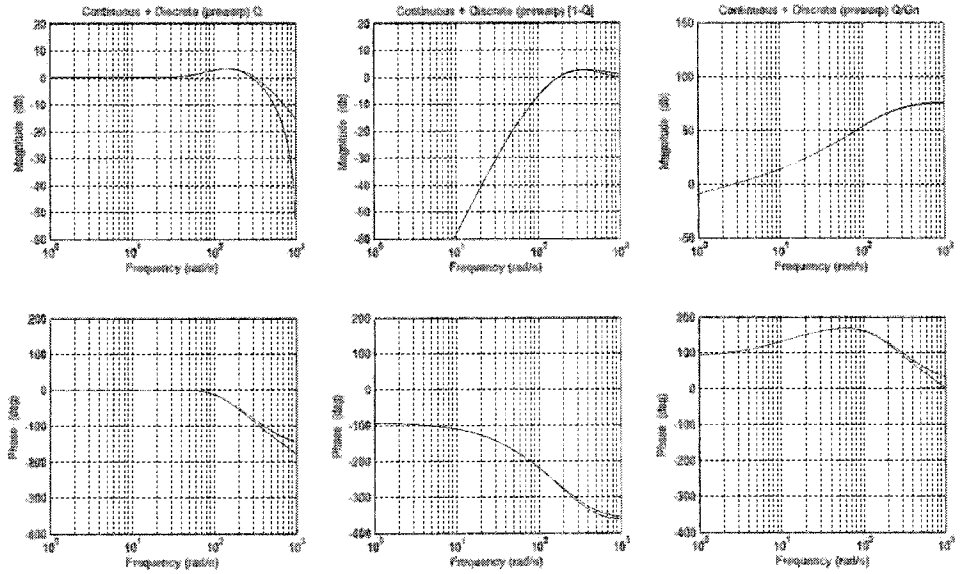

Figure C.7: **Bode Plots - Continuous vs. Discrete using Prewarp**
4^{th} Order $Q_{42D}(s)$ DOB
f_c = 68 Hz; prewarp at 17.645 rad/sec

Appendix D

dSpace© and Control Desk

ControlDesk© is the software (Figure D.1) supplied by dSpace© to interface between the operator, Simulink, and the DSP ds1102 board. Below is a description of the setup procedure to begin a test session or experiment as it is referred to by dSpace©.

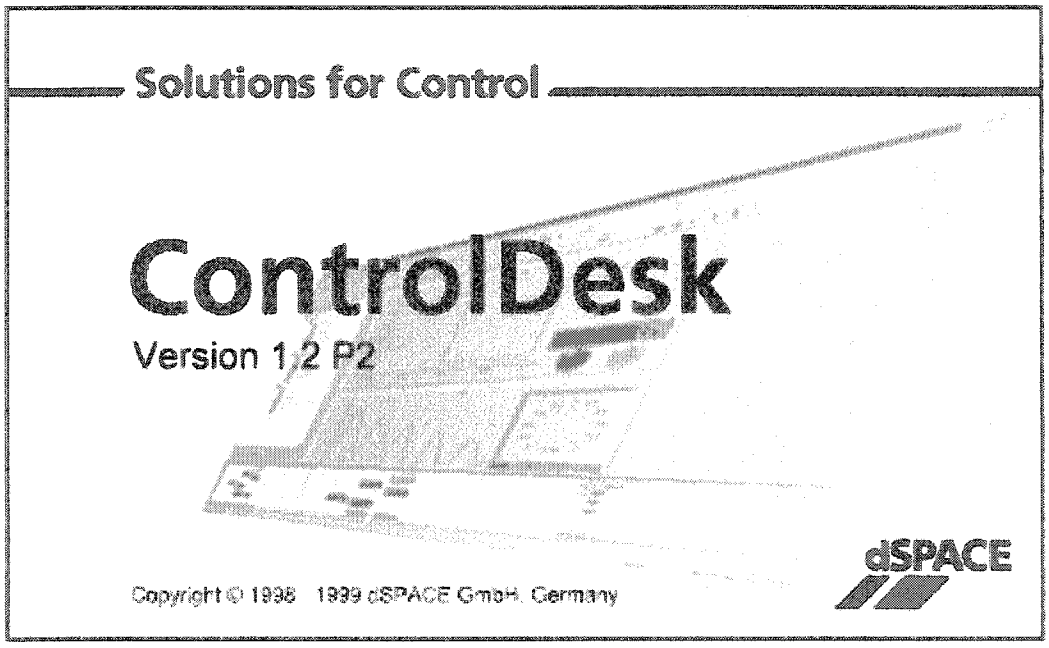

Figure D.1: ControlDesk© Splash Screen

Typically software manuals are basically worthless in communicating how to do relatively simple things. The dSpace© manual is no exception. The steps below are intended to get a new user up and running in a relatively short time. There are no doubt other ways to accomplish the end result, perhaps even better ways;

however this procedure always worked, whereas some variations did not. Below is a no-frills, step by step procedure to start an experiment. It assumes that a Simulink file and associated setup files; i.e., Matlab m-file(s) exist somewhere on the computer that is intended to be downloaded to the DSP board.

- Plug in the ribbon cable to the DSP board on the back of the computer. Plug in the two amplifier cords to the wall outlet.
- In the working directory under Matlab → Work (such as: Matlab → Work → Harlow → implementation → Experiments) create a new subdirectory (titled for example: 14Mar (use the actual date of the experiment). Do this for each new working day in the lab. Copy the Simulink model and Matlab support files to this directory. In addition, copy a file called Layout1.lay to this directory.
- Start Matlab and Simulink as normal.
- Set the Matlab path: File → Set Path...

  Under Browse: navigate to the newly created 05Mar14 directory

  Under Set Path: navigate to the newly created 05Mar14 directory

  Note: This works for the existing version of Matlab on the lab computer (R11). Other versions are likely to be somewhat different.
- Load the setup files necessary to run the Simulink model. File → Open should start in the 05Mar14 directory. If it did not, the path setup did not go

well. This is important, since all of the dSpace© files will land in this directory and there are a lot of them.

- Start dSpace© ControlDesk© using the shortcut on the desktop. Create a new experiment: File → New Experiment. A popup screen, Figure D.2, will come up. Fill in the Experiment Name (same as the working folder) and the Authors as shown. Check the Working root path (ControlDesk© puts this informationin automatically). Then click OK.

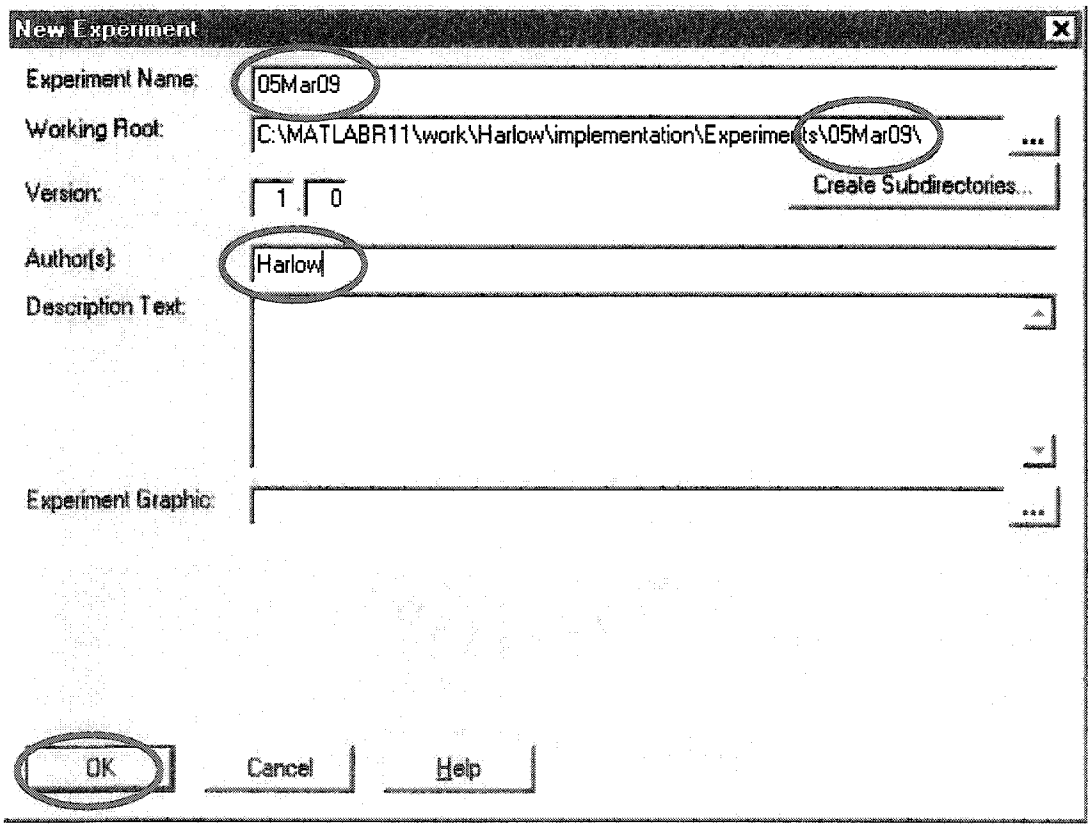

Figure D.2: ControlDesk© New Experiment Screen

- Load Layout: File → Open… the popup will be blank at first but choose file type: Layouts from the pull down menu as shown, then single click on the Layout1.lay file to select it. Then click Open.

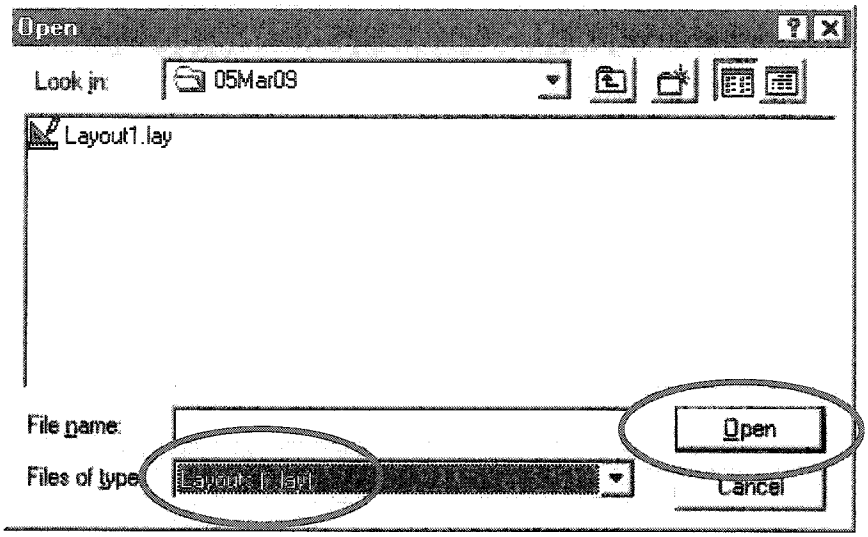

Figure D.3: ControlDesk© "Open Layout" Screen

- The experiment layout will be displayed. If a previously created layout was chosen then it might appear as in Figure D.4. Otherwise a blank layout will appear and new instruments will have to be selected from the instrument selector. This is quite easy and also addictive. There is a tremendous temptation to build the most impressive control panel the world has ever seen. However, keep it simple: just a couple of plots and the control area. Detail can always be added. The problem is that these instruments are memory hogs. Too many will prevent the experiment from operating

properly (unless the computer has 2Gb RAM). There is a small triangle and a square near the center of the menu bar. If the small square is red and the triangle is grey then an experiment from Simulink is already running. Click the red square to stop the run. The triangle should change to green.

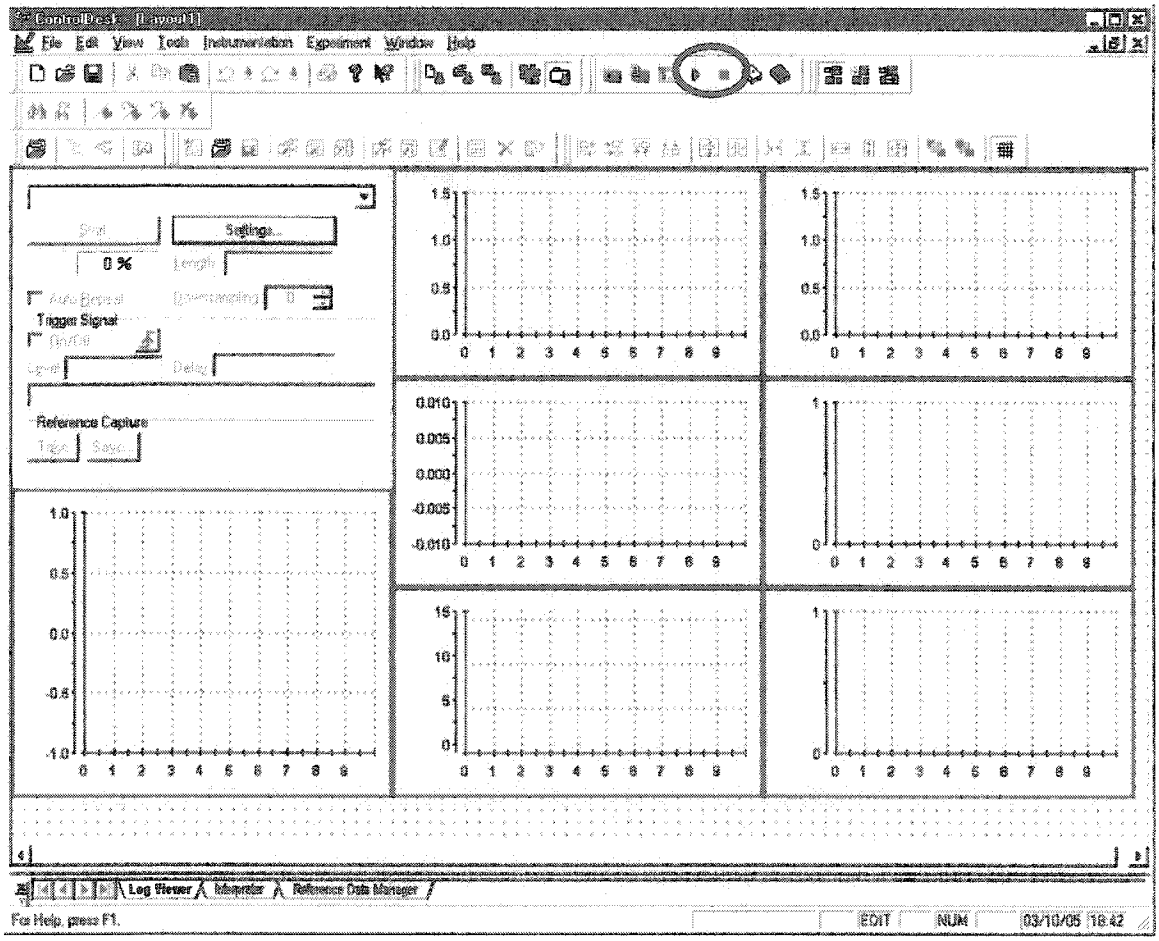

Figure D.4: ControlDesk© Layout Screen Example

- Assuming a layout similar to Figure D.4 has been loaded, go back to the Simulink model. Select Simulation → Simulation Parameters. On the popup screen there are several parameters to check: the solver must be set to "Fixed Step" and "Discrete – No Continuous States". The set the stop time to some number like 10 seconds, and set the Fixed step size to the sample time (0.003sec).
- Finally select the last tab on the same popup as above which should be Real Time Workshop. On this screen is the Build button which will take the Simulink file and translate it to C-code and finally compile it into object code. Click on the Build button.
- The Matlab command window will automatically come to the front so that any build errors can be read. If there are none, then control will be turned over to ControlDesk© as indicated by the small red square discussed earlier. Click the small red square to stop the experiment. Note: there is a small home-made black box with two switches: one of these activates the shoulder the other activates the elbow motor. The switch also reverses the sign of the current to the motor depending on which way it is toggled. For safety concerns, this switch should be centered – off – when the build button is clicked.
- Now click on the new tab at the bottom of the ControlDesk© layout that contains a lengthy path and ends with a .trc. This pane can be

expanded vertically to temporarily reveal its contents. In the left half of the pane, select Model Root. In the right half select the parameters from the Simulink block diagram and drag each one to whichever plotter is desired. Repeat until all plotters have at least one parameter associated, Figure D.5. The red outline around each layout item will then disappear. Reduce the vertical size of this pane back to nothing.

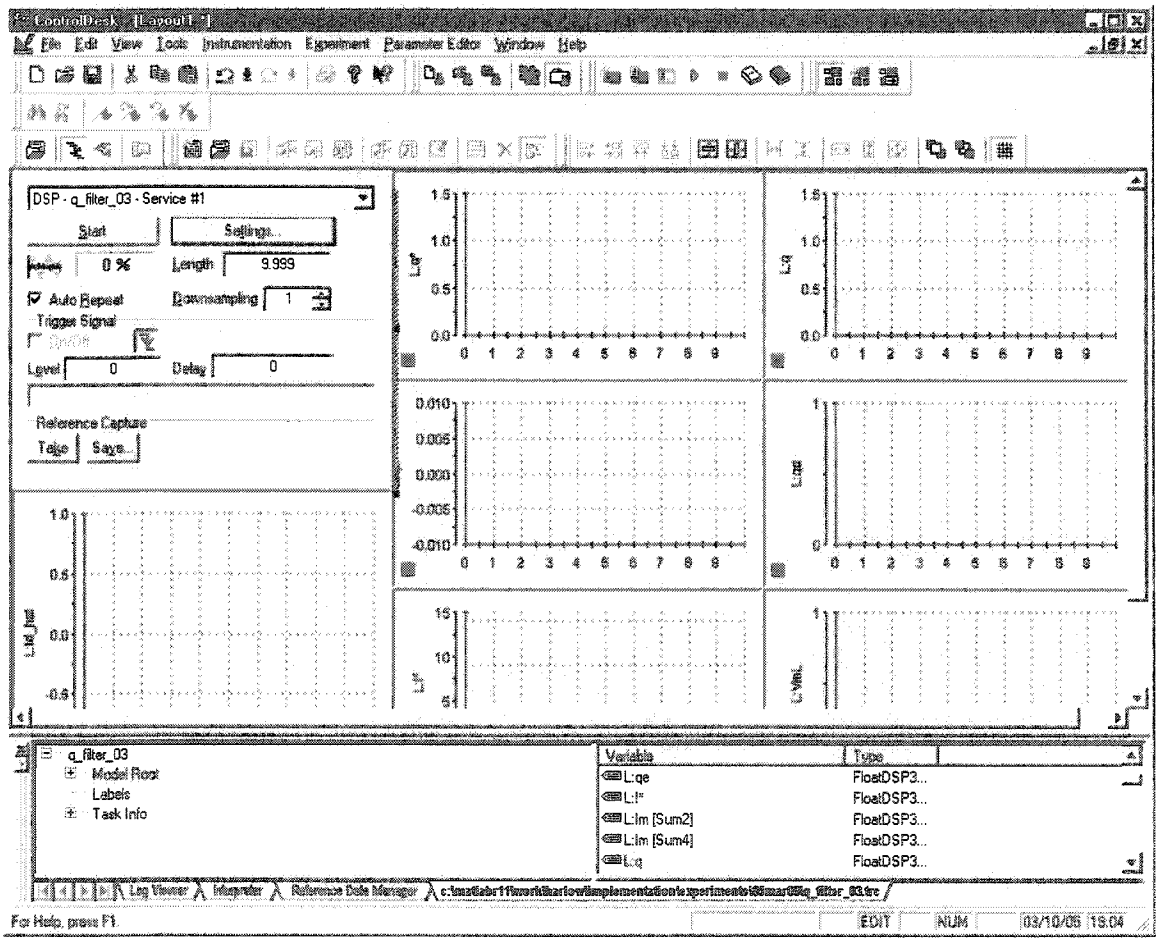

Figure D.5: ControlDesk© with .trc (Trace) Layout Screen Showing

- Note that there is currently no "Hardware" menu item. To remedy this, select View → Instrument panel. Three vertical panes open up. The right most pane is the Hardware pane. Click on the tab for this pane. Now close all three panes. There should now be a "Hardware" menu item.
- Before a run can be made, one thing remains. The run control panel on the layout (the complicated mass of buttons, and icons) must be dealt with. Double-click on this panel. A popup comes up as shown in Figure D.6. Pull down the "Capture" combo-box and set it to the Simulink model currently loaded. In most cases there will only be one item to choose from. Also set the Interval length to the same time as the Stop Time in Simulink. For some reason, when 10 sec is entered, 9.999 seconds is displayed. Click OK. Leave everything else set to their default values.

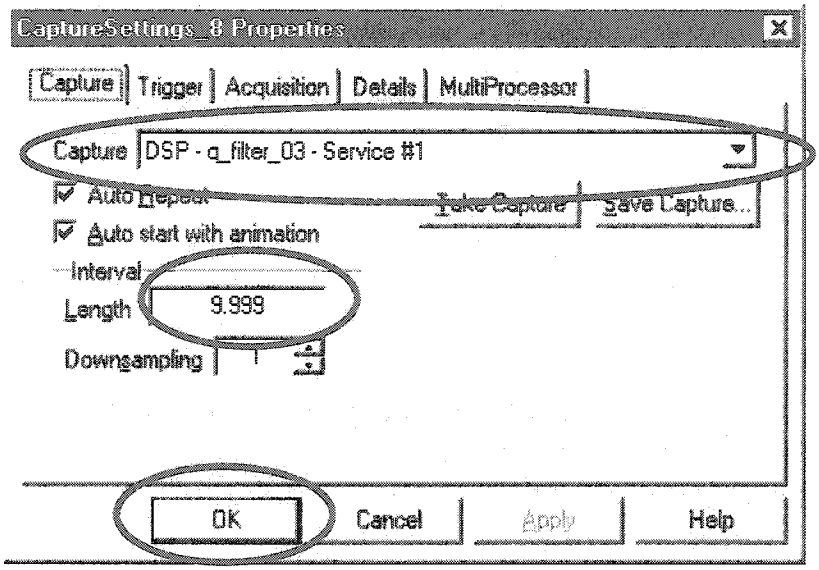

Figure D.6 ControlDesk© Capture Settings Screen

- Now time to start the run: select Hardware → Stop Real Time Process. Then again select Hardware → Load Application and select the object code file there and click open. Set robot arm to its initial position.

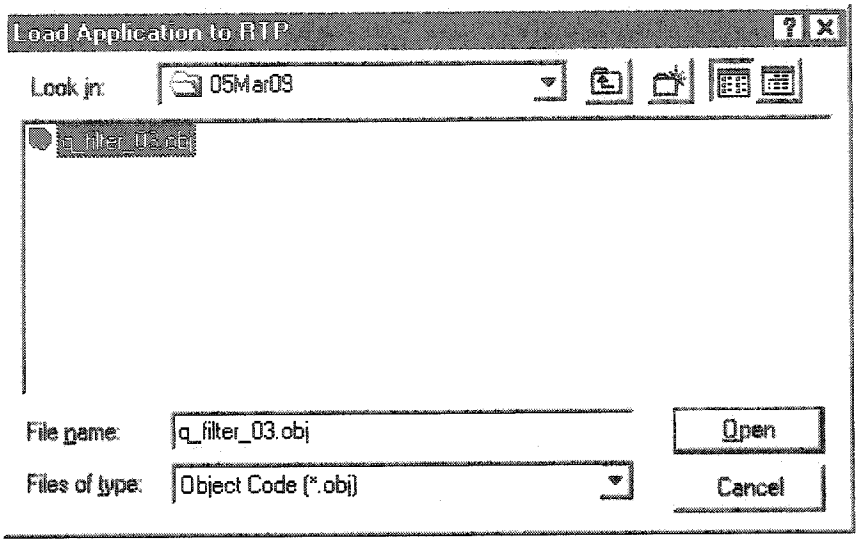

Figure D.6: ControlDesk© Object Code Load Screen

- It must be noted that there are other menu items in much too close a proximity to the previous two having to do with registering the ds1102 board. Clicking on these must be avoided at all costs. They will reset the board and without the licensing information it may not be possible to get it going again. If one of these is clicked by accident there is a cancel button, but these should not be relied on too much. Mouse clicks become far too automatic especially on late nights.
- These steps are vital: An hourglass appears indicating some time is elapsing while the object code is loaded. During this time (2-3 seconds) set the cursor on the experiment "Animation Start" icon (as shown in Figure D.7) and at the same time be ready to flick the small switch (elbow) on the

small, black homemade box (motor safety switch). When the cursor changes from hourglass to arrow, click the animation start button and flick the safety switch simultaneously.

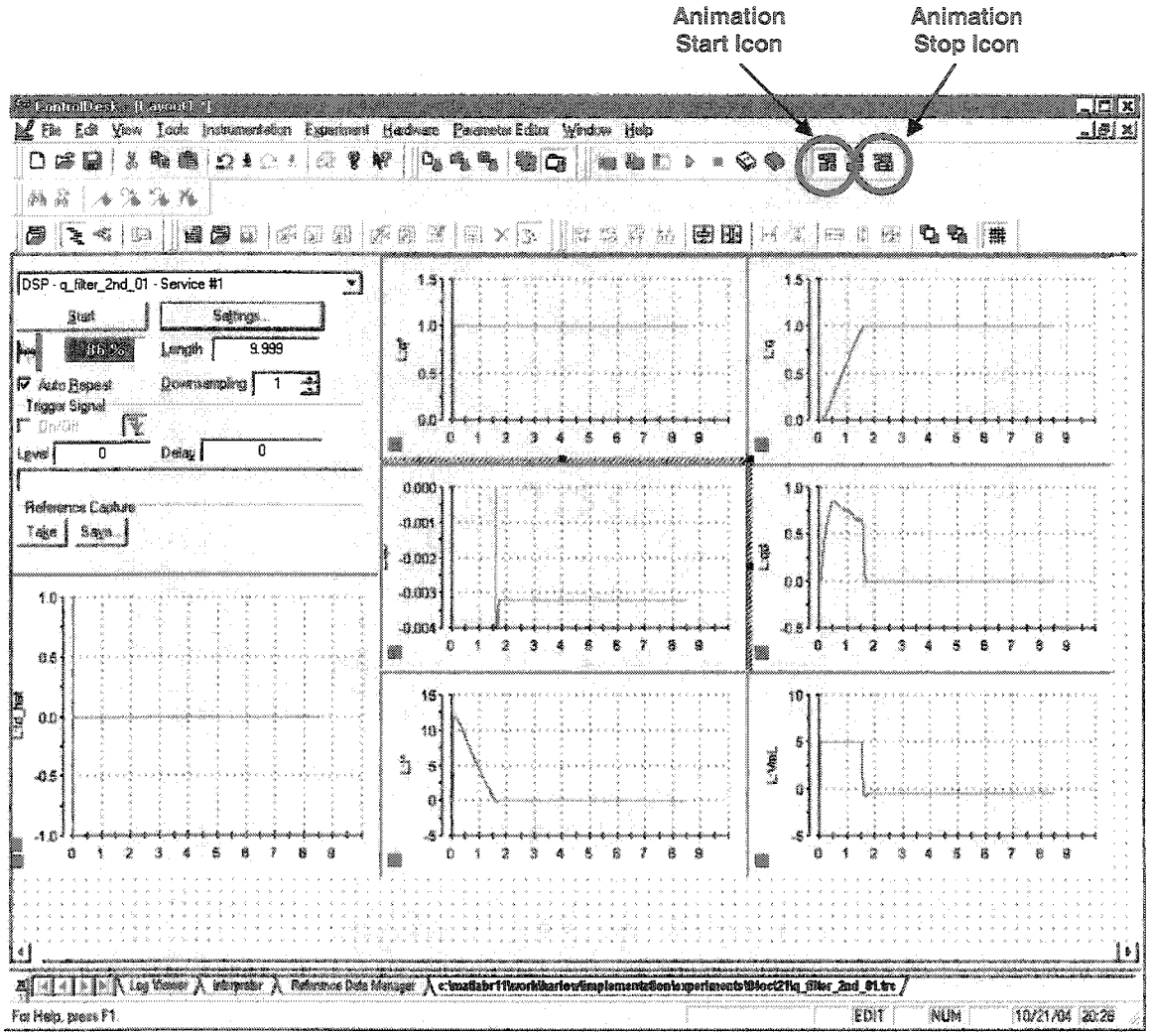

Figure D.7: ControlDesk© Experiment Screen